OUTLIER DETECTION AND MULTICOLLINEARITY IN SEQUENTIAL VARIABLE SELECTION: A LEAST ANGLE REGRESSION-BASED APPROACH

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Kelly Meredith Kirtland

January 2017

OUTLIER DETECTION AND MULTICOLLINEARITY IN SEQUENTIAL VARIABLE SELECTION:
A LEAST ANGLE REGRESSION-BASED APPROACH

Kelly Meredith Kirtland, Ph.D.
Cornell University 2017

As lasso regression has grown exceedingly popular as a tool for coping with variable selection in high-dimensional data, diagnostic methods have not kept pace. The primary difficulty of outlier detection in high-dimensional data is the inability to examine all subspaces, either simultaneously or sequentially. I explore the impact of outliers on lasso variable selection and penalty parameter estimation, and propose a tree-like outlier nominator based on the LARS algorithm. The least angle regression outlier nomination (LARON) algorithm follows variable selection paths and prediction summaries for the original data set and data subsets after removing potential outliers. This provides visual insight into the effect of specific points on lasso fits while allowing for a data-directed exploration of various subspaces.

Simulation studies indicate that LARON is generally more powerful at detecting outliers than standard diagnostics applied to Lasso models after fitting a model. One reason for this improvement is that observations with unusually high influence can inflate the penalty parameter and result in a severely underfit model. We explore this result through simulations and theoretically using a Lasso homotopy adapted for online observations. Additionally, LARON is able to explore multiple subspaces while post-hoc diagnostics rely on a variable selection that has already occurred under possible influence of an unusual obser-

vation. However, LARON underperforms random nomination when attempting to detect high leverage, non-influential points located in minor eigenvalue directions in high dimensional settings. The lack of detection appears to result from a robustness in Lasso's variable selection process against such points.

A new R package implementing the LARON algorithm is presented and its functionality to detect multicollinearity in the data, even when masked by high leverage points, described. This package is then used to analyze data created by simulation and several real data sets.

BIOGRAPHICAL SKETCH

Academia has surrounded Kelly Kirtland ever since she was a little girl growing up near Hamilton College in Clinton, NY. Although she had intended to dedicate her life to a career in music, she developed a love for math during her time at Wheaton College in Illinois. Kelly was introduced to statistics and computer programming in the summer before her senior year when she attended the Summer Institute for Training in Biostatistics sponsored by North Carolina State University and the Duke Clinical Research Institute. Despite an unbearably hot southern summer, the staunch northerner found statistics so interesting that she decided to pursue a graduate degree in the field. She was accepted into the M.S./Ph.D. program at Cornell University and began her studies immediately after completing her B.S. in Mathematics.

At Cornell she met her husband Joe who, despite being a physicist, managed to sweep her off her feet through his patience and good humor. After their first year of marriage, Joe received a job at Alfred University, causing the Kirtlands moved two hours away from Ithaca just as Kelly was beginning a new research project under Dr. Paul Velleman. After a year working exclusively on her thesis, Kelly was offered an adjunct position at Alfred University while attending a local game night; a year later, she applied for and was awarded a visiting professor position in the Division of Mathematics and Computer Science, which she has maintained and will continue to hold after the completion of her degree.

With the completion of her Ph.D. in January of 2017, Kelly fully intends to dive back into the hobbies she loves and has missed sorely during the many years since she began her degree: singing and playing the piano, losing to her husband at tennis, hiking in the Adirondacks, playing board games with friends, embroidery, traveling, staying involved in her Orthodox Christian

church, and spending time with the people she loves.

Glory to God for all things.

ACKNOWLEDGEMENTS

I would never have been able to complete this degree without the help and support of many of the Cornell University faculty and staff. Marty Wells was foundational in seeing the form of this thesis somewhere in my scattered ideas. Jacob Bien provided essential insight into the Lasso, recommending papers and discussing ideas. Giles Hooker, Bea Johnson, and Diana Drake and Bea Johnson helped me to navigate the sea of inexplicable forms and allowing me to work *in absentia*.

Thank you to my advisor Paul Velleman, who made all of this possible. Paul gave me a chance to continue my degree when few others would have undertaken the task. He was a constant fount of optimism and much needed perspective. He and his wife Sue opened to me their hearts and their home, always taking an interest in me as a person beyond my work.

My family and friend's unswerving support has been amazing during the many, many years I have remained in school. Mom and Dad, thank you for your love and encouragement, for never losing faith in me, and being proud of me no matter what. Thank you to the rest of my wonderful family, St. John the Baptist Orthodox Church, and the many, many dear friends too numerous to name who have fed me, called me, prayed for me, and given me a reason to get away from the computer and out of the house at least three times each year.

And finally, thank you to my husband Joe who continues to be the best thing to have come out of my graduate school experience. I could never have made it through without your constant support, empathy, love, and patience. You have always been there as a voice of encouragement, reason, and faith, my best friend and better half. I love you with every particle of my being!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

High dimensional data (commonly referred to as "p greater than n") introduce many unique challenges to the data analyst. All analyses focus on at least one of the following interrelated goals: variable selection, model interpretation, and prediction. It is necessary to assume that any relevant information exists only in a low-dimensional subspace, and discovery of this subspace, i.e. variable selection, should be of primary importance. Only once this has been completed is it possible to perform model estimation for the purposes of interpretation. It does not make sense to interpret coefficient values if the important variables are omitted from the subspace or their effects are swamped by unimportant variables. Optimal internal prediction may be achieved with alternative subspaces, but external prediction may suffer without correct variable selection. For example, the "noise accumulation" from including too many extraneous variables or utilizing a subspace of a higher dimension than is optimal tends to decrease prediction accuracy (see [21] and [23]).

The impact of outlying data points is frequently acknowledged in standard regression fits. Due to model-fitting tendencies to reduce prediction error, some points may pull the model towards themselves, decreasing the residual at that point and making the data point appear ordinary; this self-justifying behavior can make it quite difficult to detect an outlier numerically, which is frequently the only option in multivariate regression. There are two broad categories of outlier detection methodologies: they may be called "direct" and "indirect", as

in [30], or "diagnosis" and "accommodation" in [35]. Direct methods are largely concerned with performing diagnostics that measure the change to a model or estimator conditioned on the removal of a subset of observations. Indirect methods prefer to utilize models that are robust against unusual observations (i.e. models that are difficult for a small set of points to "bend"), and use the results of these models as the yardstick for unusual behavior.

Models intended for variable selection, particularly those in the high-dimensional case, provide little in the way of robustness against outlying observations. Indeed, due to the vast emptiness of the data space, there is a sense in which *every* observation is unusual. Subset selection, for example, was specifically cited by Tibshirani in [62] as susceptible to changes in model selection and prediction accuracy due to "small changes in the data". Ridge regression, while in a sense robust against influential observations (by down-weighting points unusual in low-eigenvalue directions; see [46]), does not provide sparse variable selection. The LAD-Lasso method proposed by Wang, Li, and Jiang [67] is only robust against heavy-tailed residuals rather than influential points. Therefore a direct approach appears to be more suitable than indirect methods. Unfortunately, known diagnostic measures can only be calculated on a low-dimensional subspace. As it is generally impossible to perform exhaustive subspace searches, this subspace is generally selected using a non-robust model fit to a potentially contaminated data set. Therefore a common mode of model diagnostic in this context involves utilizing an indirect approach (i.e. build a model using all observations, then compare observations to the model) with a model that is not sufficiently stable for the purpose.

The Lasso model is an ℓ_1-penalized regression model that is frequently used

in multiple regression scenarios with sparse coefficient vectors. It is widely used in datasets from the fields of genetics and compressive sensing, among others. There have been suggestions for adaptations of the Lasso in order to simultaneously perform outlier detection, however these require either unreasonable conditions (such as knowing a priori that a significant subset of the the observations are not outliers, as in [64]) or estimation of an additional penalty parameter. The first is clearly not suitable in common model fitting problems, and the second is likely an issue since outliers can influence data-driven estimates of penalty parameters, discussed further in Section 2.2.1.

Instead, I have examined the computation methods for determining a Lasso fit. Estimating a lasso fit relies on algorithms that sequentially or iteratively update the coefficient estimates and the set of variables active in the model. Influential observations can influence a single step in the process and the algorithm may never be able to recover. This error may even be compounded in subsequent iterations. Therefore it is important to examine the impact of observations during each step in the process.

The algorithm proposed here attempts to do just that. It incorporates standard lower-dimensional diagnostic measures into every step of the LARS algorithm for computing the Lasso. LARS, as with other Lasso algorithms, produces a sequence of possible models indexed on the amount of penalization applied to the coefficients. Thus I propose building alternative paths whenever an observation or set of observations would lead to utilizing an alternative subspace in the model. Implementing sequential diagnostics in this manner provides the following benefits:

1. Examining the influence of specific observations on the variable selection

process.

2. Allowing the subspace on which diagnostics will be performed to vary dynamically.
3. Observing the improvement in prediction accuracy through the removal of certain observations.

1.1.1 Curse of Dimensionality

Keough and Mueen [39] provide an excellent overview of the "curse of dimensionality". The phrase has been used to connote different things, but one important interpretation of this idea is the decentralization of space as the number of dimensions increases, as exemplified by taking the ratio of the volumes of the unit sphere and the unit square. As the number of dimensions goes up, the unit sphere contains an insignificant amount of volume relative to the unit square (i.e. the majority of space moves further and further from the center). In standard modeling procedures it also may be used to refer to the exponentially larger number of cases required to achieve equivalent accuracy.

A major implication of this in the context of outlier detection in high-dimensional data is in the form of measurement. Distance does not generalize intuitively into higher dimensions; there is a sense in which all points in space are considered equally far apart. This can be a problem when it comes to outlier detection since it is common to define outliers as points which are "far" (according to some metric) from the main body of data. With high dimensional data, however, it doesn't particularly make sense to utilize this idea of distance. Another approach is to incorporate some idea of "influence" on an estimator

instead. This is not necessarily better, as all observations exert a great deal of influence over models. The removal of a single point from any high-dimensional data set (however clean) may result in chaotic behavior in subset selection; the Lasso, elastic net, and ridge regression are considered more stable.

1.2 Outliers

There are several inherent challenges with outlier detection in general. Visual detection, the most effective method, is rarely possible for $p > 3$. Outliers tend to be self-justifying because they generate parameter estimates that conform to the anomalous values, so the examination of recalculated statistics after the removal of these points is usually necessary before their influence can be detected. Outlying points may also exist in groups called microclusters [53], requiring that the entire microcluster be removed before the influence of any individual point can be identified.

The detection of outliers is an essential step toward reaching the ultimate goal of every analyst: better understanding of the data and its implications for the population of interest. The bending of a model fit to a disproportionately small number of observations can result in poor predictive capabilities for new observations and incorrect interpretations of coefficient estimates. Locating unusual observations in multivariate space may also be an end in itself, such as in determining credit card fraud. This type of analysis may also be called "anomaly" detection. Outlier detection allows analysts to correct or omit values that are known to be incorrect prior to use in estimation or inferential procedures, identify cases which do not belong to the population of interest, or

discover a previously unforeseen subgroup of the population which may open new and interesting areas of research.

The causes of outliers in data are so diverse that there is no way to prescribe a general solution to their existence. If any outliers are noticed, the analyst must carefully explore the specific nature of the extraordinary point, fix any identifiable errors, and consider the modeling goals before determining a course of action. One should not be hasty and automatically remove any observation that seems to counteract a perceived trend. It is also important to recognize that there are certainly scenarios which warrant such removal. It is more common to hear invectives against the former, though the latter may cause similar problems. I have personally heard an illustrative story from a biologist who was studying the movements of fish. During the experiment, one of the electronic sensors attached to a fish malfunctioned and caused the fish to have a seizure. This one observation altered his conclusions, but he was not allowed to remove it for publishing.

1.2.1 Characteristics

The term "outliers" is one for which there is not a single agreed-upon definition. In the vernacular and in common statistical practice, it is generally intended to convey the idea presented by Grubbs [29]: "An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs." Another similar definition was proposed somewhat more recently by Bendle, Barnett & Lewis [6]: "An observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data."

However, an interpretation of the term in the context of linear models was presented by Chatterjee and Hadi [12]: "an observation for which the studentized residual is large in magnitude compared to other observations in the data set."; that is, an observation which does not follow the general model fit to the data set. I will use the term *outlier* in the sense of its more general definition, and discuss Chatterjee and Hadi's interpretation simply as a particular *characteristic* that may be exhibited by an outlier.

There are two main qualities which may be exhibited by an outlier:

1. *High Leverage*. Observations with high leverage are unusual in the predictor space and are defined identically to multivariate space ouliers as above. Regression estimates are highly susceptible to changes in the response value of these points.
2. *Large Residual*. In this context, the large residual refers to an extraordinary deviation from the "true" model, i.e. the model which best fits the majority of observations. (In practice, this point may not have an unusually large observed residual from the model fit to the full set of data.

Given these two characteristics, I will categorize outliers using the following three terms:

1. *High leverage point.* Refers to an observation with high leverage, but that follows the general trend of the data relative to the response (i.e. high leverage, small true residual).
2. *High residual point.* An observation that is not unusual in the predictors, but does not follow the general trend of the data relative to the response (i.e. low leverage, large residual).

3. *Influential point.* An observation that is *both* unusual in the predictors and does not follow the general trend of the data relative to the response (i.e. high leverage, large residual).

Chatterjee and Hadi [12] raise an excellent point that the term "influential" denotes that this one observation exerts greater impact on the results than other points; however, it is important to address the question "Influence on which results?" An observation, if omitted, may drastically change the estimates (or variances) of model coefficients, predicted values, goodness-of-fit statistics, and/or (in some cases) variable selection. For example, in a set of data where the predictors are uncorrelated with the response, a single point that is extreme in the predictors can almost uniquely determine the model coefficient estimates. In this extreme case, goodness-of-fit statistics will be excessively positive, but predictions (within the main body of data) may not be largely affected.

1.2.2 Diagnostic Measures and Procedures

Diagnostic measures have been well-explored for ordinary least squares regression nearly since its inception. These topics may be found in greater detail in any thorough book on regression and diagnostics; I am largely indebted to [12], [5], and [24].

Consider data with response vector $\mathbf{y} \in \mathbb{R}^n$ and design matrix of predictors $\mathbf{X} \in \mathbb{R}^{n \times p}$ (with columns standardized to have mean 0 and variance 1) with the following relationship:

$$\mathbf{y} = \mu + \mathbf{X}\beta + \sigma\epsilon$$

where ϵ is a vector of iid standard normal errors, μ is an intercept term, and β is a

p-dimensional sparse vector (i.e. the majority of its elements are zero). Assume that $\mathbf{X}$ may be partitioned into columns, denoted by capital letters X_j, $j = 1 \dots p$, or into rows denoted by boldface, lowercase letters $\mathbf{x}_i, i = 1 \dots n$. The least squares estimate of β is found through the minimization of the squared error loss:

$$\beta^{LS} = \arg\min_{(\mu,\beta)} \|\mathbf{y} - \mu - \mathbf{X}\beta\|_2.$$

Least squares diagnostics examine functions of the observed values in $\mathbf{y}$ and $\mathbf{X}$, the estimated coefficients β^{LS}, the predicted values $\hat{\mathbf{y}} = \mathbf{X}\beta^{LS}$, and the residuals $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$ in order to determine if the data satisfy necessary assumptions of the model and that no small subset of observations exert undue influence over the estimation procedure.

The diagonal elements of the hat matrix $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$, denoted h_i for $i = 1 \dots n$, define the leverage scores for each observation. There are multiple suggestions for cutoffs to identify unusually "high" leverage in an observation. Huber [37] used an interpretation of $\frac{1}{h_i}$ as the equivalent number of observations that determine the response of that observation $\hat{y}_i$ to suggest that any $h_i > 0.2$ (i.e. any point whose predicted value can be determined by fewer than 5 observations) should be investigated. Alternatively, since $\sum_i h_i = p$ (for a full-rank $n \times p$ matrix), equal distribution among all observations would suggest that $h_i \approx \frac{p}{n} \forall i$, any observation for which $h_i > \frac{2p}{n}$ is suspect. A third option uses the fact that, assuming a linear model with normal errors,

$$\frac{h_i - \frac{1}{n}}{1 - h_i} \sim kF_{(p-1,n-p)}$$

for known constant k to suggest a ratio of F quantiles to determine a cutoff with a given probability.

The majority of outlier detection algorithms in high-dimensional space fo-

cus only on finding points occupying an especially sparse part of the predictor space. In the context of model fitting, leverage is rather an examination of an observation's potential impact on a model. It is important to note that it is impossible in high-dimensional data to examine the entire predictor space since all standard influence measures are dependent upon the inverse of the covariance matrix, which is rank-deficient in high dimensions. Traditional measures of depth and distance have relatively little meaning in high dimensions. Angiulli [2] instead proposes using a "weight" measure summing the distance of an observation from the k-nearest neighbors. Other methods use the fact that outliers tend to exhibit high leverage only in a projection onto some lower-dimensional subspace of the predictor space. Exhaustive searches of all subspaces is rarely practicable. One alternative suggested by Aggarwal and Yu [1] uses an evolutionary approach based on subject density within grids of subspaces.

The studentized residual is the common measure to check for high residual points. Although the standard residual $e_i = \mathbf{y}_i - \hat{y}_i$ measures the actual distance from the model fit, it fails to account for the fact that most data tend to become more sparse in the tails, and thus estimators tend to exhibit higher variability. The purpose of this measure is to account for the expected increase in variability near the periphery of data points by using the leverage statistic and the standard deviation of the prediction errors σ to scale the residual. The studentized residual is defined to be

$$t_i = \frac{e_i}{\sigma\sqrt{1-h_i}}.$$

Since this measure depends on the unknown standard deviation of the errors, it is necessary to estimate it using the (potentially contaminated) data available. Two ways of calculating this value lead to different versions of the studentized residual: the internal and the external. These are sometimes also referred to

as standardized and studentized residuals respectively. Internally studentized residuals utilize the usual estimate of residual variance including all observations

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{n} e_k^2}{n-p}$$

while externally studentized residuals utilize the "leave-one-out" estimate $\hat{\sigma}^2_{(i)}$, or the variance of all residuals except the one under investigation. In cases where $n \leq p$ leading all leverages h_i to be equivalently 1 (or even when p is close to n, so all leverage scores are nearly 1), neither of these measures is practicable. Therefore it has become more common to simply look at the residual values $\mathbf{e}$ themselves, or standardized only by some estimate of σ.

Single row deletion diagnostics are by far the most common mode of examining data for influential observations. The DFBETAS statistics examine the impact of a single observation on the coefficient estimates of a linear regression β^{LS} relative to their variance

$$\text{DFBETAS}_{ij} = \frac{\beta_j^{LS} - \beta_{(i),j}^{LS}}{\hat{\sigma}_{(i)} \sqrt{(\mathbf{X}^T\mathbf{X})^{-1}_{ii}}}$$

where A_{ii} represents the i^{th} diagonal element of matrix A, β_j^{LS} represents the j^{th} element of β^{LS}, and the subscript (i) indicates that the i^{th} observation was omitted prior to estimation. This, for location estimators, reduces to

$$\frac{e_i \sqrt{n}}{\hat{\sigma}_{(i)} (n-1)}.$$

Chatterjee and Hadi [12] recommend that observations with $|\text{DFBETAS}_i| > \frac{2}{n}$ be examined further as possible outliers. DFFITS is an alternative that measures the change in the fitted value from omitting one observation. It takes the value

$$\text{DFFITS}_i = \frac{\mathbf{x}_i \left(\beta^{LS} - \beta_{(i)}^{LS}\right)}{\hat{\sigma}_{(i)} \sqrt{h_i}} = \frac{e_i \sqrt{h_i}}{\hat{\sigma}_{(i)} (1-h_i)}$$

It is sufficient to look only at the change in fit for the observation being omitted as all other changes will be less than DFFITS$_i$ in absolute value. A cutoff suggested by [5] is $2\sqrt{\frac{p}{n}}$. Cook's distance is probably the most common of all leave-one-out diagnostics, likely due to its distributional underpinnings. Cook's distance is intuitively identical to DFBETAS, however instead of springboarding off of the *externally* studentized residual, it utilizes instead the *internally* studentized residual. Although this sacrifices some of the extra detection capabilities of DFBETAS by utilizing a potentially inflated variance estimate, the ability to obtain confidence ellipsoids and probabilities is undoubtedly a worthy advantage. Cook's distance is given by

$$D_i = \frac{\left(\beta^{LS} - \beta^{LS}_{(i)}\right)^T \left(\mathbf{X}^T\mathbf{X}\right)\left(\beta^{LS} - \beta^{LS}_{(i)}\right)}{p\hat{\sigma}^2} = \frac{t_i^2 h_i}{p\,(1-h_i)}$$

where t_i is the i^{th} internally studentized residual. A cutoff of $\frac{4}{n-p-1}$ has been suggested. Cook suggests, under the assumption of normal errors,comparing D_i to the $F_{p,n-p}$ distribution for "descriptive levels of significance" (quoted in [12]).Although there is no exact distributional equivalence, this evolved from the fact that the comparable formula comparing $\left(\beta - \beta^{LS}\right)$ does follow the given F distribution.

High dimensional data analysis is sufficiently new that there do not appear to be any standard measures of influence. The current influence measures just described will clearly not apply since, as p approaches n, all h_i values will approach 1 and all influence measures will approach infinity. Often, reliance is placed on robustness of high-dimensional model building methods; for example, ridge regression is known to down-weight the leverage of a given observation in minor eigenvalue directions more than in major eigenvalue directions (see Lichtenstein [46]). However, it is uncertain whether this robustness is truly

desirable in high-dimensional data, even if it could be proven for other modeling techniques. Certainly, it is an advantage to have robustness against any point that is influential due to error or to non-inclusion in the population of interest. However, at least in the present state of this field, every observation is obtained at high cost. If a single unusual observation is the sole representative of an important subgroup of the population, one would not want to see the effects of that observation omitted or diminished. Thus it seems particularly advisable in the high-dimensional setting that strong preliminary analysis and data cleaning are more advantageous than robust model-building techniques.

Current methods for high-dimensional data focus only on "unusualness" within predictor space rather than effects on a model fit. This work is intended to focus specifically on measuring the influence of observations on the three elements of interest in the lasso model: coefficient estimates (for interpretation), prediction, and sparsity.

1.3 Multicollinearity

Multicollinearity (aka collinearity) is a feature of data where a subset of predictors are related through a linear relationship. This may be exhibited through an exact linear relationship (e.g. weight measured in both pounds and kilograms) or linear relationships with relatively small errors (e.g. GPA and SAT scores). Perfect collinearity can cause singular (thus non-invertible) covariance matrices; highly collinear design matrices are often considered "nearly" invertible and may produce large errors when inverted. Common ways to determine whether data are collinear are by looking for large pairwise correlations, examining the

variance inflation factors of each variable, and calculating the condition number of the covariance matrix. The variance inflation factor (VIF) for variable j is defined to be

$$\text{VIF}_j = \frac{1}{1 - R_j^2}$$

where R_j^2 is the R^2 value obtained by regressing variable X_j on all other covariates. Typically, a variable is considered to be collinear if the VIF is larger than 5 or 10. The condition number

$$\kappa = \sqrt{\frac{\lambda_1}{\lambda_p}}$$

looks at the invertibility of the covariance matrix $\mathbf{X}^T\mathbf{X}$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ are its eigenvalues. A common rule is that $\kappa > 30$ indicates multicollinearity, however this value may be determined by the amount of round-off error deemed acceptable in the calculations. See [57] and [34] for a more in-depth discussion of this issue.

1.4 Lasso Model

The "least absolute shrinkage and selection operator" (hereafter the "Lasso") model is an ℓ_1 penalized regression which enforces sparsity in the coefficient estimates. A general penalized regression estimation problem with arbitrary penalty function $P(\beta)$ and loss function L may be formulated with a constraint, as in

$$\hat{\beta} = \arg\min_{\beta} L(\beta \big| \mathbf{X}, \mathbf{y}) \qquad \text{subject to } P(\beta) \leq t$$

or in a Lagrangian form

$$\hat{\beta} = \arg\min_{\beta} \left\{ L(\beta \big| \mathbf{X}, \mathbf{y}) + \lambda P(\beta) \right\}.$$

Ridge regression ([48], [18]) utilizing squared-error loss $L(\beta|\mathbf{X}, \mathbf{y}) = \sum_i \{y_i - \mathbf{x}_i\beta\}^2$ and an ℓ_2-norm penalty $P(\beta) = \sum_j \beta_j^2$, was one of the first penalized regressions to be widely used. Although able to produce models capable of good prediction in high-dimensional contexts, it fails to provide interpretable models as it does not perform any variable selection or produce sparse coefficient estimates.

The Lasso was originally proposed by Tibshirani [62] to maintain the gain in prediction accuracy that ridge had obtained over ordinary least squares (OLS) regression while also improving interpretability. The first is obtained by introducing a small amount of bias in exchange for a decrease in the large variances, the second by focusing only on those variables with the strongest effects on the response.

The lasso estimates are defined to be

$$\left(\hat{\mu}, \hat{\beta}\right) = \underset{(\mu,\beta)}{\arg\min} \left\{ \sum_{i=1}^{n} (y_i - \mu - \mathbf{x}_i\beta)^2 \right\} \qquad \text{subject to } \|\beta\|_1 \leq t$$

where $\|\mathbf{a}\|_1 = \sum_j |a_j|$ and t is some pre-determined penalty limit. This may alternatively be expressed in its Lagrangian form

$$\left(\hat{\mu}, \hat{\beta}\right) = \underset{(\mu,\beta)}{\arg\min} \left\{ \sum_{i=1}^{n} (y_i - \mu - \mathbf{x}_i\beta)^2 \right\} + \lambda\|\beta\|_1$$

As μ is unpenalized, it is clear that $\hat{\mu} \equiv \bar{y}$, so we may assume that y has been centered without loss of generality. In practice, it is necessary to account for the estimation of μ in the degrees of freedom estimation, however we will assume that $\hat{y} = 0$ *a priori* except in the simulations and examples of Chapters 3 and 4.

The advantage of the ℓ_1-norm penalty function is that it forces many coefficients to be exactly 0. Unlike the ridge, which performs shrinkage on the coefficient estimates uniformly on a ball within the coefficient space, the Lasso

forces estimates falling below the residual error to 0 and shrinks the remaining estimates uniformly on a diamond.

1.4.1 Bayes Perspective

The Bayes formulation equivalent to the Lasso under the assumption of normal errors was first given in the original Lasso paper by Tibshirani [62] and has been further explored by Park and Casella [54] and Strawderman, Wells, and Schifano [60], among others. Due to the strongly data-driven methods for determining the penalty parameter (discussed in Section 2.2), the Bayes viewpoint has considerable appeal.

The lasso estimates are equivalent to the posterior mode of the Bayes model expressed hierarchically:

$$\begin{aligned} \mathbf{y} \,|\, \mathbf{X}, \beta, \sigma^2 &\sim \mathbf{N}_n \left(\mathbf{X}\beta, \sigma^2 \mathbf{I}_n \right) \\ \beta_j \,|\, \lambda &\overset{iid}{\sim} \text{DoubExp}\,(\tau) \end{aligned}$$

where $A \sim$DoubExp(τ) implies that A has density

$$\pi(a) = \frac{1}{2\tau} \exp\left\{ \frac{-|a|}{\tau} \right\}$$

with $\tau = \frac{1}{\lambda}$ (the inverse Lagrangian penalty parameter) is estimated using standard Bayesian methods such as marginal maximum likelihood. There are several alternative models that have also been presented. Representation of the double exponential as an inverse-gamma scaled mixture of normals may be used to add a level to the hierarchy as in [60] and allow for easier sampling. Others have suggested using the ratio $\frac{\sigma}{\lambda}$ as a scale-invariant alternative hyperparameter to the double exponential distribution, which ensures a unimodal

full posterior [54].

An additional benefit of Bayesian methods is the ability to apply a hyper-prior to λ, which is unlikely to be a truly "fixed" value. An appropriate hyper-prior should be fairly flat, ensure positivity, and approach 0 sufficiently quickly as $\lambda \to \infty$. Park and Casella [54] recommend a class of gamma priors on λ^2 to guarantee positive λ, to maintain a proper posterior, and because of its easy conjugacy. Thus the prior on lambda takes the form

$$\pi\left(\lambda^2\right) = \frac{\delta^r}{\Gamma(r)}\left(\lambda^2\right)^{r-1}\exp\left\{-\delta\lambda^2\right\}.$$

1.4.2 Properties and Optimality Conditions

A significant challenge in any theoretical study of the lasso algorithm is the unknown penalty parameter, and thus all theories rest on the assumption that λ is fixed appropriately *a priori*. Although not particularly believable, the assumption is necessary in order to obtain results concerning consistency and detection. Additionally, there is generally some assumption restricting the amount of collinearity that can exist between the features corresponding to true zero coefficients and those with non-zero coefficients. This may be difficult to achieve in some cases, such as with unfiltered genetic data. Overviews of these properties may be obtained in [23] and [32].

The Karush-Kuhn-Tucker (KKT) conditions are a general title given to conditions that guarantee the existence of an optimal solution to a non-linear programming problem such as the Lasso. Broadly, they ensure that the problem has the following qualities:

1. *Stationarity*: if a solution exists, then the differential of the cost function must equal zero somewhere.
2. *Primal and dual feasibility*: it is possible to satisfy all conditions in the primal and dual of the problem simultaneously.
3. *Complementary slackness*: any solution with slack (or equality) in the primal must have equality (or slack) in the dual, and vice versa.

The KKT conditions specific to the loss function of the Lasso are satisfied when

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\beta}) = \lambda\gamma$$

where $\gamma \in \mathbb{R}^p$ is some subgradient of the ℓ_1-norm evaluated at $\hat{\beta}$ with elements of the form

$$\gamma_j \in \begin{cases} \operatorname{sign}\left(\hat{\beta}_j\right) & \text{if } \beta_j \neq 0 \\ [-1, 1] & \text{if } \beta_j = 0. \end{cases}$$

Satisfying this equation guarantees the existence of an optimal Lasso solution. These conditions provide a basis for most of the theoretical results concerning the Lasso.

Various studies have shown that the Lasso exhibits sign consistency of the estimator, and thus is able to successfully identify the correct active set, if the underlying model is truly sparse and under some additional restrictions on the predictor matrix $\mathbf{X}$ (for example, see [50] and [10]). However, it is also known that the coefficient estimates themselves are not consistent. It is therefore common to utilize an additional method such as a repetition of the Lasso (aka. relaxed Lasso, [49]), or treat the Lasso as a variable selection method and then perform ordinary least squares to obtain estimates.

The univariate Lasso coefficient estimates are equivalent to the soft-thresholding operator of Donoho and Johnstone [17]

$$\hat{\beta}_j = \text{sign}\left(\hat{\beta}_j^{LS}\right)\left(\left|\hat{\beta}_j^{LS}\right| - \gamma\right)_+$$

where $\hat{\beta}^{LS}$ is the ordinary least squares estimate, $(\cdot)_+$ returns the argument if positive and 0 otherwise, and γ is selected so that $\|\beta\|_1 \leq t$ is satisfied. Therefore all results concerning this operator (e.g. that $\hat{\beta}_j = 0$ if $\beta_j < \sigma$) also hold for the Lasso.

The Dantzig selector (DS) is an alternative ℓ_1-penalized regression method similar to the Lasso, though replacing traditional squared-error loss with the ℓ_∞-norm. Under the condition that $\|\beta\|_1 \leq t$, the Lasso and DS are asymptotically equivalent as $t \to \infty$. Thus any asymptotic results that hold for DS also hold for the Lasso. The solution to DS is simply a linear programming problem, and its risk follows the oracle estimator proportional to to $\sqrt{\frac{2 \log p}{n}}$ under certain conditions [23]. Similarly, bounds can be placed on the prediction error for the Lasso depending on the assumptions in place for the design matrix. These bounds are classified as fast- or slow-rate depending on how quickly the prediction error converges to 0, generally as a function of the dimension of the predictor matrix and the residual variance. Fast-rate type bounds (such as those in [16]) generally require a restricted eigenvalue condition; slow-rate bounds leave the design matrix mostly unrestricted but then require a much larger sample size in order to obtain a similarly accurate model (see [33]). These bounds may be used to determine the penalty parameter λ in lieu of more data-driven methods.

Fan and Li [22] established the conditions necessary to have a sparse and asymptotically normal optimizer with probability one, also known as the weak

oracle property for nonconvex penalized likelihoods. Although the Lasso estimator does not satisfy this property due to bias, the adaptive Lasso was suggested by Zou [71] to address the excess bias and does satisfy the oracle property under some additional regularity conditions.

1.4.3 Penalty Parameter Forms

There are multiple ways in which the ℓ_1 penalty function may be constrained. Thus far, a constraint $||\beta||_1 \leq t$ and the addition of $\lambda||\beta||_1$ to the minimization function have been introduced. In the Bayesian context, the penalty is the hyperparameter for the double exponential prior on the coefficients. The amount of sparsity is therefore controlled by the selection of the penalty parameters (or, using a broader term, tuning parameters) t, λ, or τ. These values are assumed to be fixed in theory, however in practice they are selected through data-driven methods such as cross-validation, Empirical Bayes, or according to some function of data characteristics such as sample size, number of covariates, and estimated residual variance. The constraint and Lagrange forms are related in the sense that there are values of each which will produce identical lasso fits on the same dataset.

The Lagrangian penalty parameter is a unitless value that is generally assumed to balance the sparsity of the resulting model and the residual variance [27], which makes it difficult to interpret in the context of multiple datasets. It is additionally difficult to determine whether or not the selected value of λ is reasonable. Although the constraint formulation is generally more interpretable, it is still difficult to compare the effects of different values of t used for differ-

ent models. To account for the relative effects of different data, it is common (particularly when using the LARS algorithm described in Section 2.3) to use a fractional form of the penalty parameter. This new constraint is expressed

$$\frac{\left\|\hat{\beta}_t^{lasso}\right\|_1}{\left\|\hat{\beta}^{full}\right\|_1} \leq s$$

where $\hat{\beta}^{full}$ is the coefficient from the full Lasso model, after the maximum number of predictors, $\min(n, p)$, have been added to the active set and $\hat{\beta}_t^{lasso}$ is the Lasso estimate subject to the constraint $\|\beta\| \leq t$.

A third penalization form is to limit the number of non-zero elements of β. Let $\mathcal{A} = \left\{j : \beta_j \neq 0\right\}$ be termed the active set of predictors included in the model, and let $|\mathcal{A}|$ denote the cardinality of set $\mathcal{A}$. Thus a constraint could be imposed such that $|\mathcal{A}| < \varkappa$. Models with this type of constraint are not necessarily equivalent to a Lasso solution since they derive from a model with a penalty function of the form $P(\beta) = \sum_j I\left\{|\beta_j| > 0\right\}$; however, the addition of this constraint may be incorporated as a method for limiting the selection of penalty parameters in order to produce models consistent levels of sparsity under different data conditions.

CHAPTER 2

LASSO ESTIMATION

2.1 Overview of Estimation Methods

Hebiri and Lederer [33] define three broad classes of Lasso estimation methods: homotopy methods, interior-point methods, and "shooting" algorithms. The two most common approaches utilize the coordinate descent (CD) algorithm proposed in [26], a type of "shooting" algorithm, and the homotopic least-angle regression (LARS) algorithm of [20]. Hastie, Tibshirani and Friedman describe the LARS as a "'democratic' version of forward-stagewise regression" [32]; it adds variables to the model "one-at-a-time" in a similar manner, however it does not calculate the full regression model for each subset. More information on the LARS may be found in Section 2.3. Alternative homotopy methods have also been proposed (e.g. [52] and [47]).

Coordinate descent has become much more popular than the LARS, especially with the introduction of the `glmnet` R package, which relies solely on the Lagrangian form of the problem. For every λ over some grid, coefficient estimates are updated cyclically until the optimal values are reached. Its computational efficiency is two-fold: the use of a "warm start", i.e. using the optimal estimates from the previous λ value as initial values for the next step, and the soft-threshold operator for univariate coefficient estimates. Other shooting algorithms have been proposed by others, such as [68] and [65]. Saha and Tewari [58] discuss the nonasymptotic convergence and dominance that such cyclic procedures have over gradient descent methods.

Interior point methods have been proposed by [13] and [41]. Standard convex optimization methods may also be leveraged, for example in conjunction with quadratic programming in [7] and [3].

2.2 Selection of Penalty Parameters

The most common mode of selecting penalty parameters is through the use of k-fold cross-validation (CV). An excellent overview of CV is available in chapter 7 of [32]. The goal of CV is to estimate the prediction error, and usually its variance as well. Creating a model and then testing its prediction error on the set used to generate the model yields a smaller error (known as in-sample error) than testing against independently-drawn data (or out-of-sample error). A simple solution is to split the data set into two independent groups, use one set (the "training" set) to create the model and the other (the "validation" set) to assess the out-of-sample error. This is an excellent method if the data set is large, though in high dimensional data where data is scarce it is infeasible. CV utilizes a series of small-scale training-validition divisions: first partition the data set randomly into k groups, and sequentially assign one group at a time to be the validation set; after fitting a model to the remaining sets, calculate the prediction error on the selected validation set; repeat the previous step until all partitions have been used for validation once. The results may then be used to calculate the mean (CV error) and standard error. The selection of k is important as it balances bias and variance considerations. CV is essentially unbiased as k approaches n (aka leave-one-out CV), but tends to have an inflated variance since all training sets are so similar; as k decreases, the variance is lowered but bias increases.

Generally, the object in fitting a model is to minimize the amount of prediction error; with the Lasso in particular, the model fit will be unique for a given penalty parameter, say λ. Therefore it is the selection of the penalty parameter that gives the Lasso its ability to achieve better prediction. To determine the appropriate parameter value, perform CV on the data over some sufficiently fine grid of λ values, often beginning with the smallest value such that a null model is selected, and decreasing incrementally to 0. The resulting estimates may be used to determine the optimal value.

Common sense would suggest selecting a value for λ that produces the minimum CV error, however this method is not particularly stable and fails to provide optimal sparsity. Figure 2.1 provides sample plots from two CV processes. In the image on the right, there is little noticeable change in the error produced by models associated with $\log(\lambda)$ values between -3 and -0.5, and there is no stable way to predict that the true minimum average will occur at one point along this "plateau" rather than another; in fact, a small change in the grid increment or a different seed for the random number generator may be the most notable explanation for selecting one value over another. Thus it is more advisable to implement the 1SE rule of Breiman et al. [8]: add together the minimum CV error and one standard error associated with the same model and select the λ value associated with the sparsest model with a lower CV error. This increases the stability of the prediction and tends to choose sparser models.

There are some who rely instead upon the theoretical bounds on prediction error mentioned in Section 2.2 to select the penalty parameter. Generally these bounds are only applicable if the selected λ is large enough to obtain a high probability of selecting the true active set. One such option is the so-called "uni-

versal choice" of $\lambda = \sqrt{\frac{2\log(p)}{n}}$. These values obviate the need for any potentially time-consuming use of CV, however they are not as adaptable to variations in the dataset. [16] discuss a bound on the prediction error given this selection if the true active set is sufficiently small relative to n, but show that mild correlations can significantly decrease the rate of convergence.

Lederer and Müller in [43] aptly describe the pervasiveness of methodologies requiring a tuning parameter in high-dimensional variable selection procedures, and the need for a "tuning parameter that is properly adjusted to all aspects of the model [which is] difficult to calibrate in practice." They also decry the use of CV to select λ as "computationally inefficient" and producing "unsatisfactory variable selection performance". The poor variable selection performance of the CV usually comes from using the model associated with the minimum RSS which tends to overfit the model; implementing the 1SE rule, while generally mitigating the under-penalization issue, vastly overcorrects in the face of unusual observations. It is important to note, however, that both CV-selection methods can be affected by even moderately influential observations.

Ryan Tibshirani and Jonathan Taylor [63] discuss ways to view the degrees of freedom of a Lasso problem in terms of the variable selection, i.e. the "effective number of parameters". They discuss choosing tuning parameters based on Mallow's C_p as a computationally-efficient alternative to CV. However, this begs the following question: how do they determine the estimated degrees of freedom? They address Stein's Unbiased Risk Estimate (SURE) in relation to degrees of freedom, where

$$df(g) = E\left[(\nabla \cdot g)(y)\right]$$

where $\nabla \cdot g$ is the divergence of g, some estimator. The difficulty of both unbiased

estimators, though, involves either knowing the model fits from the unknown model or knowing the expected value of the predictions.

2.2.1 Outliers and the Penalty Parameter

Outliers can exert undue influence over the coefficient estimates and the active set selection of a Lasso fit. However, outliers also can affect the amount of sparsity induced in the model. This influence is most notable in CV-selected penalty parameters, even in low-dimensional settings. Interestingly, the two methods of selecting an appropriate penalty parameter (i.e. the minimum and 1SE rules) appear to have opposite reactions to the presence of outliers.

A preliminary study in order to examine the effect of outliers on the CV-selected Lasso penalty parameter involved generating a data set with 100 observations on 12 Gaussian, uncorrelated predictors. The response was modeled using the standard linear equation $\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon$ where ϵ is standard normal, $\sigma = 7$, and $\beta = (5, -5, 5, -5, 0, \ldots, 0)^T$. Figure 2.2 shows the resulting impact of taking a single observation and moving it away from the rest of the data set in either the predictor space, the residual dimension, or both. Instead of documenting the change to the penalty parameter λ which is not directly comparable between different data sets, the size of the active set and the average in-sample prediction error provide a clearer examination of the effect of the selected penalty parameter on the model fit. The in-sample error is useful (rather than the CV error) as it reflects how closely the selected model follows the outlying point.

The instability of the minimum rule is clearly evident in the additional variability in the curves over the 1SE rule, especially in the residual plot (the middle

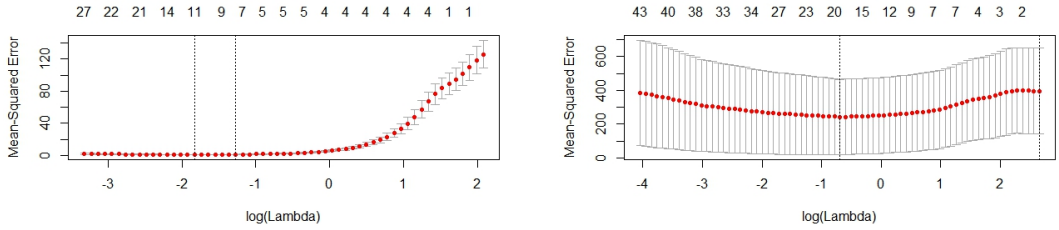

Figure 2.1: Plots representing the CV error for various values of λ. The dashed lines represent the minimum and 1SE selected penalty parameters. The figure on the left included only clean observations; the figure on the right with one influential point.

image in the left column of Figure 2.2). This instability is a direct result of the minimum occurring arbitrarily within the wide range of models with statistically equivalent CV errors (the plateau in Figure 2.1). It is also evident that the minimum rule tends to favor less-sparse models with in order to achieve the best possible prediction accuracy, while the 1SE rule tends to exhange more interpretable, sparse models for some prediction accuracy. This trade-off is well-known, and should be carefully considered in the context of a given analytic goal.

In the presence of outliers, the minimum rule tends to select much larger models than if the outlying observation were to be removed. In Figure 2.1, the minimum CV error shifts far to the right with the introduction of an outlier, however the model sizes associated with each λ value have become significantly larger; despite the larger penalty, the minimum rule would select a model with nearly 20 predictors instead of 12. The 1SE rule, however, tends to select more sparse models due to the vast increase in the CV estimated standard errors. In the CV plots above, the largest standard error for a clean dataset had a value of approximately 20, or 15% of the estimated CV errors; with the outlier, the smallest standard error is approximately 200, or 66% of the estimated CV error.

This leads to an excessive reduction in the size of the active set selected, and tends toward a null model.

According to Figure 2.2, leverage does not appear to have much effect on the size of the active set except that the size of the model associated with the minimum rule becomes increasingly volatile as h_i approaches 1; the 1SE models are unaffected. Although it is not always evident due to the scale of the plots (which were set thusly for comparison purposes), 1SE models have consistently higher prediction errors and the prediction error becomes more volatile for both methods as leverage increases. The behavior is the same whether the leverage occurs in major or minor eigenvector directions, however the effect is more pronounced when along major eigenvector directions.

With a high residual outlier, the 1SE rule proves to be quite stable, selecting the correct number of features for the model over a wide range of residual values; however if the observation is pushed sufficiently far from the model (at approximately 15σ), the penalty parameter quickly grows to force a uniform selection of the null model. The minimum rule model has no such dramatic drop to the null, however there does appear to be a steadily slow decline in the active set size. This persistence in a high variable selection rate is thus able to maintain astonishingly good in-sample prediction accuracy, which is also a warning that that the model tends to follow the outlying point, despite its relatively low leverage ($h_i \approx 0.25$).

The most intriguing graphs in Figure 2.2 is in the selected size of the active set for influential observations. In every other occurrence, an increase in observation distance was met by either a decrease in active set size or indifference, but the two methods tended to agree in the direction if not in the magnitude or

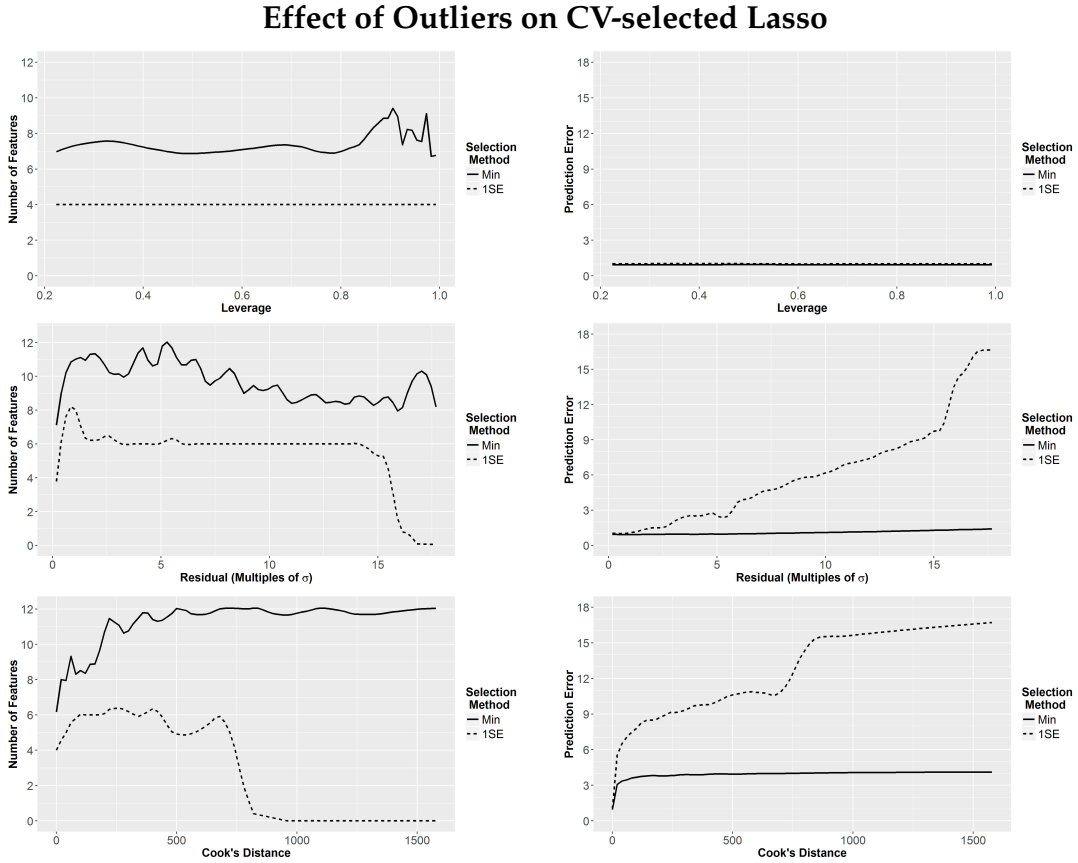

Figure 2.2: Change in active set selection (left column) and average prediction error (right column) of CV-selected Lasso models in the presence of outliers. Uncorrelated data set with $n = 100$ and $p = 12$, with a true active set of four predictors. Outliers introduced as (from top to bottom): high leverage points, high residual points, and influential points.

rapidity of the effect. Here, however, we see entirely opposite results based on the penalty parameter selection method: the 1SE rule model tends toward the null model as influence increases as before, however the minimum rule model tends toward the full model.

In higher dimensional situations, the minimum rule appears to pick increasingly larger models both for high residual points and influential points, though it levels out before the full model is reached. In a situation with 50 observations and 45 variables, the plot plateaued between 25 and 30 features for the minimum model in the presence of either high residual or influential points. The

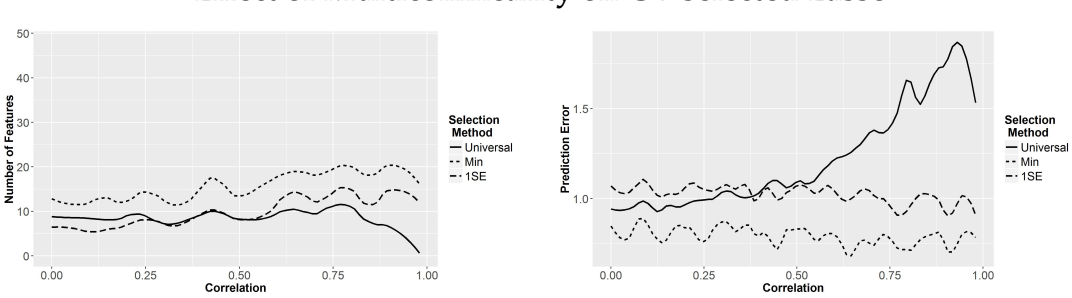

Figure 2.3: Change in active set selection (left column) and average prediction error (right column) to CV-selected Lasso model in the presence of autoregressive multicollinearity. Data generated from a multivariate normal with $n = 50$, $p = 45$, and an active set of size 4.

1SE model behaved in a similar manner to the low-dimensional setting, though it selected a larger number of predictors for a short time before dropping to the null model.

2.2.2 Multicollinearity and the Penalty Parameter

As correlation between predictors increases, it also appears as though cross-validation tends to select larger models in the presence of higher correlation between predictors. This trend is similar for both selection methods, with the minimum rule consistently selecting on average five more predictors than the 1SE rule.

Hebiri and Lederer [33] discuss the fact that penalty parameters based on fast-rate and slow-rate bounds (which depend only on sample size, number of parameters, and variance) tend to suffer when it comes to prediction accuracy. CV tends to work better as a method of parameter selection in the face of highly collinear predictors. This seems to be clearly exhibited in Figure 2.3, where the "universal" penalty parameter $\lambda = \frac{2\log(p)}{n}$ tends to under-select the model. How-

ever, as concluded in [33], even in these instances some artificial inflation of the penalty parameter (inducing greater sparsity) would improve model interpretability at the expense of some prediction bias.

2.3 Least Angle Regression

Consider the usual regression data structure with $n \times p$ predictor matrix $\mathbf{X}$ with columns standardized to have mean 0 and variance 1 and response vector $\mathbf{y}$ centered at 0. The true underlying model is assumed to follow the sparse regression form

$$\mathbf{y} = \mathbf{X}\beta + \sigma^2\epsilon$$

where $\epsilon_i \overset{iid}{\sim} N(0,1)$ and most elements of β are zero. Let $\mathcal{A}$ be the *active* set, the set of indices of all non-zero elements of β, with cardinality $|\mathcal{A}|$. There are three primary possible purposes for fitting the model: identifying the variables that are members of the active set, interpreting the coefficient estimate $\hat{\beta}$, and obtaining the predicted values $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$. The model is to be fitted according to the Lasso model discussed in Section 1.4.

The LARS algorithm described in [20] for estimating the Lasso is a (near) homotopy which introduces variables sequentially into the model. The underlying principal is that the estimated coefficient vector is updated through an adaptive piecewise-linear function (therefore differentiable almost everywhere) on a gradually increasing subset of predictors and responses, with nodes occurring whenever new variables enter the model.

As penalty parameter values are generally unknown a priori, the LARS algorithm (as with many other Lasso estimation methods) operates in a similar

way to standard tree-building methods like CART [8]: begin with a null model and gradually add in variables until the model is "full". This model path may then be "trimmed" to an optimal size by setting the penalty parameter to an appropriate value.

2.3.1 Geometric Motivation

The geometric idea underlying the LARS algorithm is that only those variables most correlated with the residuals should be included in the model. Imagine that the coefficient estimate vector traces out a piecewise linear path through the coordinate space parametrically indexed on the penalty fraction $s \in [0, 1]$ described in Section 2.2, where $s = 0$ corresponds to the null model where $\mathcal{A} = \emptyset$ and $s = 1$ corresponds to the full regression model. The regression model is considered "full" when $\mathcal{A}$ contains either all variable indices or there are no more available degrees of freedom (i.e. $|\mathcal{A}| = \min(n - 1, p)$ if the intercept is estimated).

At $s = 0$, the path starts at the origin and begins moving along the axis of the coefficient associated with the variable which is most correlated with the response (which acts as the initial residuals, since $\mathbf{y}$ may be assumed to be centered without loss of generality). At a certain point, another predictor variable yields a correlation with the corresponding residuals equal to that of the initial variable. At this point the path experiences a node (or joint, or elbow) in order to introduce the new variable into the model. The path continues in a linear fashion along the new "equiangular" direction (that is, the vector direction which bisects the angle between the previous coefficient trajectory and the new

axis). When another variable becomes equally correlated with the residuals corresponding to the point along the coefficient path, the path again experiences another node and changes direction. This continues until the model is full. For instances where $p < n$, the coefficient vector when $s = 1$ is the OLS solution.

2.3.2 Algorithm Details

This description of the LARS algorithm relies heavily on the paper by Efron et. al. [20]. Begin by centering the response vector $\mathbf{y}$, and centering and scaling the predictor matrix columns to have mean 0 and variance 1. To calculate the path, begin with a null model such that $\mathcal{A} = \emptyset$, i.e. $\hat{\beta} = 0$ and the residuals $\mathbf{e} = \mathbf{y}$. Determine correlations between the residuals and each of the predictors using $\hat{\mathbf{C}} = \mathbf{e}^T\mathbf{X}$. Let $C_{\max} = \max_j \left|\hat{c}_j\right|$ be the largest correlation in absolute value and $\hat{j} = \arg\max_{j \notin \mathcal{A}} \left|\hat{c}_j\right|$ be the index of the variable(s) $\mathbf{X}_{\hat{j}}$ most correlated with $\mathbf{e}$. To update the coefficient vector estimate, it is necessary to determine the new direction of the trajectory and also the distance along this vector to travel before the next variable should enter the model. Assume that at the current node we have obtained a coefficient estimate $\hat{\beta}_0$ with corresponding fitted values $\hat{\mathbf{y}}_0$, residuals $\mathbf{e}_0$, and correlation vector $\hat{\mathbf{C}}_0$ through initialization or completion of the previous step.

The active set $\mathcal{A}_0 = \left\{j : \hat{\beta}_{0,j} \neq 0\right\}$ consists of the indices of all variables that are included in the model corresponding to the current node. Begin by updating $\mathcal{A}$ so that $\mathcal{A} = \left\{j : \hat{c}_j = \hat{\mathbf{C}}_{\max}\right\}$. Let $\mathbf{X}_{\mathcal{A}}$ be the design matrix including only those columns with indices in $\mathcal{A}$, let $G_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}$, and let $\mathbf{1}_{\mathcal{A}}$ be a column vector of ones

Algorithm 1: LARS Algorithm for Lasso fit

Data: centered $\mathbf{y}$; centered and scaled $\mathbf{X}_{n\times p}$

Result: LARS path of coefficient vector $\beta_{\lambda'}$ indexed by penalty parameter λ' (as step or fraction)

Initialize:

- Coefficient vector $\beta = \mathbf{0}$
- Active set $\mathcal{A} = \emptyset$
- Residuals $\mathbf{e} = \mathbf{y}$
- Penalty $\lambda' = 0$

while $|\mathcal{A}| < \min(n, p)$ **do**

- $\mathbf{C} = \mathbf{e}^T\mathbf{X}$
- $\mathbf{C}_{\max} = \max \left|c_j\right|$
- $\mathcal{A} = \left\{j : \left|c_j\right| = \mathbf{C}_{\max}\right\}$
- Determine new coefficient direction
  - $G_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}$
  - $\mathbf{w}_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T G_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-1/2} G_{\mathcal{A}}^{-1}$
- Determine distance in new direction to next node
  - $\mathbf{a} = \mathbf{X}^T\mathbf{X}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}.$
  - $\mathbf{a}_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T G_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-1/2}$
  - $\hat{\gamma} = \min_{j\notin\mathcal{A}}^{+}\left\{\frac{C_{\max} - c_j}{\mathbf{a}_{\mathcal{A}} - \mathbf{a}_j}, \frac{C_{\max} + c_j}{\mathbf{a}_{\mathcal{A}} + \mathbf{a}_j}\right\}$
- Update:
  - β
  - $\mathbf{e}$
  - λ'

with order equal to $|\mathcal{A}|$. The new coefficient direction is calculated by

$$\mathbf{w}_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T G_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-\frac{1}{2}} G_{\mathcal{A}}^{-1}.$$

The new coefficient estimates are updated using the equation

$$\hat{\beta}(\gamma) = \hat{\beta}_0 + \gamma\mathbf{w}_{\mathcal{A}}$$

where γ is a scalar multiple. Now that the direction has been determined, the next step is to determine the distance (represented by γ) to the next node.

To determine this, let $\mathbf{a} = \mathbf{X}^T\mathbf{X}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}$. Note that $\mathbf{a}_{\mathcal{A}} = \left(\mathbf{1}_{\mathcal{A}}^T G_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-\frac{1}{2}}\mathbf{1}_{\mathcal{A}}$, however for convenience I will use $\mathbf{a}_{\mathcal{A}}$ to represent the scalar $\left(\mathbf{1}_{\mathcal{A}}^T G_{\mathcal{A}}^{-1}\mathbf{1}_{\mathcal{A}}\right)^{-\frac{1}{2}}$ rather than the

vector form previously given. Parametric vector functions on γ can be used to represent the new fitted values

$$\hat{\mathbf{y}}(\gamma) = \hat{\mathbf{y}}_0 + \gamma \mathbf{X}_{\mathcal{A}} \mathbf{w}_{\mathcal{A}}$$

and the correlation vector

$$C(\gamma) = \mathbf{X}^T (\mathbf{y} - \hat{\mathbf{y}}(\gamma)) = \hat{\mathbf{C}}_0 - \gamma \mathbf{a}$$

as the coefficient vector $\hat{\beta}(\gamma)$ progresses along the path defined above. Note that the elements of $C(\gamma)$ belonging to $\mathcal{A}$ will remain equivalent and largest (in absolute value) as γ changes. Let us call this value $C_{\mathcal{A}}(\gamma) = C_{\max} - \gamma \mathbf{a}_{\mathcal{A}}$. A new variable will enter the model when some $\left|c_j(\gamma)\right| = C_{\mathcal{A}}(\gamma)$ with $j \notin \mathcal{A}$. Setting the two values equal to each other and solving for γ yields the estimate

$$\hat{\gamma} = \min_{j \notin \mathcal{A}}{}^{+} \left\{ \frac{C_{\max} - c_j}{\mathbf{a}_{\mathcal{A}} - \mathbf{a}_j}, \frac{C_{\max} + c_j}{\mathbf{a}_{\mathcal{A}} + \mathbf{a}_j} \right\}$$

where $\min^+$ indicates that the minimum is only taken over positive arguments.

2.4 Homotopy for Sequential Observations

Standard least squares (LS) regression is known to produce better predictions when every predictor is added to the model, regardless of each predictor's true relationship with the response. Lasso regression is one of a family of penalized regression techniques that counterbalance this tendency in order to achieve a sparser estimate. In the Lasso, a weighted ℓ_1 penalty on the estimated parameter vector β is added to the minimization problem, i.e.

$$\hat{\beta} = \arg\min_{\beta} \|X\beta - y\|_2^2 + \mu \|\beta\|_1$$

where X is the n-by-p matrix of predictors, y is the n-dimensional vector of responses, μ is a prespecified regularization parameter, and $\|a\|_1 = \sum_i |a_i|$ is the ℓ_1-norm. There have been several proposed methods for solving this minimization problem. The most commonly used is the least angle regression (LARS) algorithm of Efron et. al. [20]. Garrigues & El Ghaoui [27] developed an iterative update homotopy algorithm specifically designed for estimates where observations are received over time.

Suppose that the $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}}$ are the data for the n observations that have already been received, and the coefficient vector $\beta^{(n)}$ and regularization parameter μ_n have already been estimated. In order to update these estimates when the new observation $(y_{n+1}, \mathbf{x}_{n+1}) \in \mathbb{R} \times \mathbb{R}^p$ (the augmented data set is denoted as X and y), Garrigues and El Ghaoui [27] presented the new minimization equation which weights the $(n+1)^{st}$ observation by t.

$$\beta(t, \mu) = \arg\min_{\beta} \left\{ \frac{1}{2} \left\| \begin{pmatrix} \tilde{\mathbf{X}} \\ t\mathbf{x}_{n+1}^T \end{pmatrix} \beta - \begin{pmatrix} \tilde{\mathbf{y}} \\ t y_{n+1} \end{pmatrix} \right\|_2^2 + \mu \|\beta\|_1 \right\}.$$

So $\beta^{(n)} = \beta(0, \mu_n)$ is known from the first n observations, and the goal is to obtain $\beta^{(n+1)} = \beta(1, \mu_{n+1})$. Let the vector $v = sign(\beta^{(n)})$ and the active set $\mathcal{A} = \{j : \beta_j^{(n)} \neq 0\}$; matrices and vectors subscripted with 1 utilize only the columns in $\mathcal{A}$. The authors propose updating the regularization parameter using

$$\mu_{n+1} = \frac{n}{n+1} \mu_n \exp\left\{ 2n\eta x_{n+1,1}^T \left(X_1^T X_1 \right)^{-1} v_1 \left(x_{n+1,1}^T \beta_1 - y_{n+1} \right) \right\}.$$

Fixing $\mu = \mu_{n+1}$, then looking at the way $\beta(t) = \beta(t, \mu_{n+1})$ changes as t varies from 0 to 1 shows the changes that occur to the active set $\mathcal{A}$ by incorporating the $(n+1)$st observation. These t's can be calculated explicitly.

Using the Sherman-Morrison formula and focusing on $\beta_1(t)$ (as all elements

not in $\mathcal{A}$ are 0), the equation can be expressed

$$\beta_1(t) = \tilde{\beta}_1 - \frac{(t^2-1)\bar{e}}{1+\alpha(t^2-1)}u$$

where

$$\begin{aligned}
\tilde{\beta}_1 &= \left(X_1^T X_1\right)^{-1}\left(X_1^T y - \mu v_1\right) \\
\bar{e} &= x_{n+1,1}^T \tilde{\beta}_1 - y_{n+1} \\
u &= \left(X_1^T X_1\right)^{-1} x_{n+1,1} \\
\alpha &= x_{n+1,1}^T u.
\end{aligned}$$

The active set can change either by having a previously non-null coefficient becoming 0, or by having a 0 element become non-null. In the first scenario, the next value of t at which component $\beta_{1i}(t), i \in \mathcal{A}$ becomes 0 is at

$$t_{1i} = \left(1 + \left(\frac{\bar{e}u_i}{\tilde{\beta}_{1i}} - \alpha\right)^{-1}\right)^{\frac{1}{2}}.$$

For the second scenario, let X_2 be the columns of X not in the active set, $\tilde{e}$ be the $(n+1)$ dimensional vector of residuals, c_j be the jth column of X_2, and $x^{(j)}$ be the jth column of $x_{n+1,2}$. The next value of t at which the jth component of $\mathcal{A}^C$ will join $\mathcal{A}$ when t is equal to

$$t_{2j} = \min \begin{cases} t_{2j}^+ &= \left(1 + \left(\frac{\bar{e}\left(x^{(j)} - c_j^T X_1 u\right)}{\mu - c_j^T \tilde{e}} - \alpha\right)^{-1}\right)^{\frac{1}{2}} \\ t_{2j}^- &= \left(1 + \left(\frac{\bar{e}\left(x^{(j)} - c_j^T X_1 u\right)}{-\mu - c_j^T \tilde{e}} - \alpha\right)^{-1}\right)^{\frac{1}{2}} \end{cases}.$$

Then the next change to the active set will occur at $t' = \min\{\min_i t_{1i}, \min_j t_{2j}\}$. If $t' \notin [0,1]$, then incorporating the $(n+1)^{st}$ observation will leave $\mathcal{A}$ unchanged; otherwise, the active set will change, so the process is repeated until $t' \notin [0,1]$.

Algorithm 2: RecLasso Algorithm

Data: $\tilde{\mathbf{X}}, \tilde{\mathbf{y}}, (y_{n+1}, \mathbf{x}_{n+1}), \mu_n, \beta^{(n)}$
Result: New coefficient vector $\beta^{(n+1)}$
Compute μ_{n+1} and $\beta(0, \mu_{n+1})$
Initialize:
  Active set $\mathcal{A}$ to non-zero elements of $\beta(0, \mu_{n+1})$
  Sign vector v of $\beta(0, \mu_{n+1})$
  $t' = 0$
while $t' \in [0, 1]$ **or** *new* $t' <$ *previous* t' **do**
  if $\beta_{1,i}(t') = 0$ **then**
    Remove i from $\mathcal{A}$
    Update v, setting $v_i = 0$
  else
    jth coefficient of $\beta(t') \neq 0$
    Add j to $\mathcal{A}$
    Update v, so $v_j = sign(\beta_j(t'))$
  Update X_1, v_1, and $x_{n+1,1}$ according to new $\mathcal{A}$
  Update $\tilde{\beta}_1 = \left(X_1^T X_1\right)^{-1} \left(X_1^T y - \mu_{n+1} v_1\right)$
  Calculate t_{1i} and t_{2j}
  new $t' = \min\left\{\min_i t_{1i}, t_{2j}\right\}$
Compute final $\tilde{\beta}_1$

This algorithm provides an excellent way to mathematically examine the influence of a single observation on the penalty parameter estimation demonstrated numerically in Section 2.2.1. First, write the standard Lasso formula in Lagrangian form such that

$$\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} \left(x_i^T \beta - y_i\right)^2 + n\lambda_n \|\beta\|_1.$$

Note that the Lagrangian penalty parameter is given in the form $n\lambda_n$, so that λ_n acts as a trade-off mechanism between the ℓ_1-norm of the coefficients and the average MSE. In the lemma below, I show that the addition of an outlying observation will tend to increase the updated penalty parameter λ_{n+1}.

Lemma 2.4.1. *Assume that a data set* $(y_i, \mathbf{x}_i)_{i=1\ldots n}$ *with n observations and p predictors satisfies all assumptions for a Lasso model. This data has been used to fit a Lasso model*

according to the minimization problem

$$\beta(\lambda_n) = \arg\min_{\beta} \frac{1}{2n} \sum_{i=1}^{n} \left(x_i^T \beta - y_i\right)^2 + n\lambda_n \|\beta\|_1$$

with an appropriately selected penalty parameter λ_n. Let $(y_{n+1}, \mathbf{x}_{n+1})$ be a new observation and its error with respect to the current model as a function of λ be

$$\eta(\lambda) = (\mathbf{x}_{n+1}\beta(\lambda_n) - y_{n+1})^2 .$$

Then if λ is updated by

$$\log \lambda_{n+1} = \log \lambda_n - \xi \frac{\partial \eta}{\partial \log \lambda}(\lambda_n)$$

so as to maintain the weighting factor between MSPE and ℓ_1-norm, the updated estimate λ_{n+1} will increase exponentially to infinity as the influence of $(y_{n+1}, \mathbf{x}_{n+1})$ goes to infinity.

Proof. Let $\mathbf{x}_i$ be the row vector of order p for the i^{th} observation, $\mathbf{X}$ be the original $n \times p$ design matrix, $\mathcal{A}$ be the active set of predictors with non-zero coefficient estimates for the model fit to the original data, and any predictor matrix of vector subscripted with $\mathcal{A}$ denote inclusion of only those columns belonging to $\mathcal{A}$. Thus the coefficient estimate for the original model (given estimate λ_n) can be characterized with the matrix equation

$$\beta(\lambda_n) = \left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1} \left(\mathbf{X}_{\mathcal{A}}^T \mathbf{y} - n\lambda_n \mathbf{v}_{\mathcal{A}}\right).$$

where $\mathbf{v}_{\mathcal{A}}$ is the vector of signs of the coefficients. This equation is substituted into the error equation $\eta(\lambda)$ and its gradient is computed. Note that the gradient is determined along $\log \lambda$ in order to ensure a positive λ.

$$\begin{aligned}
\frac{\partial \eta}{\partial \log \lambda}(\lambda) &= \frac{\partial}{\partial \log \lambda}(y_{n+1} - \mathbf{x}_{n+1}\beta(\lambda))^2 \\
&= \frac{\partial}{\partial u}\left(y_{n+1} - \mathbf{x}_{n+1}\left(\left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1}\left(X_{\mathcal{A}}^T \mathbf{y} - ne^u \mathbf{v}_{\mathcal{A}}\right)\right)\right)^2 \\
&= 2\left(y_{n+1} - \mathbf{x}_{n+1}\left(\left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1}\left(X_{\mathcal{A}}^T \mathbf{y} - n\lambda \mathbf{v}_{\mathcal{A}}\right)\right)\right) \times \\
&\qquad \mathbf{x}_{n+1}\left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1} n\lambda \mathbf{v}_{\mathcal{A}}.
\end{aligned}$$

Simplifying this expression and incorporating it into the penalty update equation yields

$$\lambda_{n+1} = \lambda_n \exp\left\{2\xi n \mathbf{x}_{n+1,\mathcal{A}} \left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1} \left(\mathbf{x}_{n+1}\beta\left(\lambda_n\right) - y_{n+1}\right)\right\}.$$

Now it is necessary to connect the exponential term to measures of influence. Although there are various ways to measure influence for the i^{th} observation, nearly all include a product of the residual $e_i = y_i - \mathbf{x}_i\hat{\beta}$ and leverage $h_i = \mathbf{x}_i \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{x}_i^T$. For example,

$$\text{DFFITS} \propto e_i \frac{\sqrt{h_i}}{1 - h_i}$$

looks quite similar to the exponential term of the new λ_{n+1} equation (Cook's distance is proportional to the square of DFFITS). To clearly see the corresponding components in the exponential term, they have been labeled below:

$$2\xi n \underbrace{\mathbf{x}_{\mathcal{A}}^{*T} \left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}}\right)^{-1} v_1}_{\propto \sqrt{h_i}} \underbrace{\left(\mathbf{x}_{n+1}\beta\left(\lambda_n\right) - y_{n+1}\right)}_{=e_{n+1}}$$

Therefore as $e_{n+1}\sqrt{h_i}$ diverges to infinity, the corresponding influence measure and the updated Lagrangian penalty λ_{n+1} will also diverge to infinity.

□

The practical impact of a penalty parameter tending to infinity is that any non-zero coefficient value will be harshly penalized, forcing selection of the null model. Although the process employed in this proof is not the same as in CV-selection, the basic view of the penalty parameter as a weighting factor to balance the mean square prediction error and the ℓ_1-norm of the coefficient vector is similar to that of the 1SE CV rule. The minimum rule, on the other

hand, tends to use λ to minimize only the error. This result confirms the observations from Section 2.2.1 that an increase in influence of a single observation will also increase the CV-selected penalty parameter. Given this result it is not surprising that an increase in leverage has no noticeable impact on the size of the active set or the prediction error in Figure 2.2, since $0 \leq h_i \leq 1$ and cannot send the exponential term to infinity, while increasing the residual or the influence eventually tend to under-fit the model. The theoretical impact of this result with finite influence is not clear since it is not possible to compare the relative impact of two different Lagrangian penalty values on the sparsity and prediction of the models fit to two different sets of data, such as by changing the value of an observation within the set. However, this effect is noticeable for observations with influence values well within the realm of possibility even in low dimensions, and is only aggravated in high dimensions when all leverage values converge to one.

CHAPTER 3

LEAST ANGLE REGRESSION OUTLIER NOMINATION

3.1 Algorithm

The least angle regression outlier nomination (LARON) process involves performing a "back-check" for points which may have influenced the variable selection at each step in the LARS process. This section will not delve deeply into the details of the LARS algorithm; for more information see Section 2.3.

Begin by assuming that there are no variables in the model, with centered and scaled design matrix $\mathbf{X}$ and centered response vector $\mathbf{y}$. Calculate the correlations between the predictors and the response

$$\mathbf{C} = \mathbf{y}^T\mathbf{X}$$

and determine the variable with the largest correlation in absolute value

$$k = \left\{ j \,:\, \left|c_j\right| = \mathbf{C}_{\max},\ j = 1 \ldots p \right\}$$

where $\mathbf{C}_{\max} = \max_j \left|c_j\right|$.

At this point, perform a standard influence measure M_i for each observation $i = 1 \ldots n$ (e.g. Cook's distance; see Section 1.2.2) assuming a linear model using all observations and variable $\mathbf{X}_k$. For some appropriate cutoff τ, collect nominated influential observation indices in the set $\mathcal{O} = \{i \,:\, M_i > \tau\}$ and form a new data set omitting these observations

$$\mathbf{X}^* = \left[x_{ij}\right], \qquad \mathbf{y}^* = [y_i] \quad \text{for} \quad i \notin \mathcal{O},\ j = 1 \ldots p.$$

Using this new dataset, recalculate correlations and most correlated variable:

$$\begin{aligned} \mathbf{C}^* &= \mathbf{y}^{*T}\mathbf{X} \\ \mathbf{C}^*_{\max} &= \max \left|c^*_j\right| \\ k^* &= \left\{j : \left|c^*_j\right| = \mathbf{C}^*_{\max},\ j = 1 \ldots p\right\} \end{aligned}$$

If $k^* \neq k$, then a split occurs: the original dataset will continue to follow the usual LARS path (in some sense the "trunk" of the resulting tree), but we will also record the path of this new branch for the remainder of its LARS path continuing from the split point.

Additional criteria are necessary in practice when it comes to the formation of new branches. As in traditional tree-building software, it is requred that there be a minimum number of cases in order to split a node. Similarly, LARON will not create a new branch if the number of observations to be included in the proposed node is less than the current size of the active set (inequality becomes inclusive if the intercept is also being modeled) in order to avoid an ill-conditioned matrix inverse.

3.1.1 Why use LARS?

It is reasonable to question why this outlier nominator should be based on the LARS algorithm rather than another estimation method. For example, the coordinate descent algorithm is much more widely used, faster, and would allow the algorithm to determine the full Lasso path for each potential subset of cases. These are indeed advantages which warrant exploration, however the LARS algorithm provides other advantages that are particularly appealing.

This algorithm arose specifically out of the desire to understand how indi-

Algorithm 3: LARON Algorithm for Outlier and Collinearity Nominations

Data: centered $\mathbf{y}$; centered and scaled $\mathbf{X}_{n\times p}$
Result: Information on nominated outliers and potential collinear variables
Initialize:
Coefficient vector $\beta_0 = \mathbf{0}$
Active set $\mathcal{A}_0 = \emptyset$
Residuals $\mathbf{e}_0 = \mathbf{y}$
Total number of branches $B = 0$
while $\exists\ |\mathcal{A}_b| < \min(n - \mathcal{O}_b, p),\ b \in 0 \ldots B$ **do**
  for $\{b\ :\ |\mathcal{A}_b| < \min(n - \mathcal{O}_b, p),\ b = 1 \ldots B\}$ **do**
    Determine new variable for b^{th} branch
      $\mathbf{C} = \mathbf{e}^T\mathbf{X}$
      $\mathbf{C}_{\max} = \max \left|c_j\right|$
      $k = \left\{j\ :\ \left|c_j\right| = \mathbf{C}_{\max} \wedge j \notin \mathcal{A}_b, j = 1 \ldots p\right\}$
    Complete LARS step for b^{th} branch with $\mathcal{A}_b = \{\mathcal{A}, k\}$
    Check for collinearity in $\mathcal{A}_b$
    Use standard influence measure to nominate cases $\mathcal{O}$
    Determine new variable without cases
      $\mathbf{C}^* = \mathbf{e}^T_{\{\mathcal{O}\}}\mathbf{X}_{\{\mathcal{O}\}}$
      $\mathbf{C}^*_{\max} = \max \left|c^*_j\right|$
      $k^* = \left\{j\ :\ \left|c^*_j\right| = \mathbf{C}^*_{\max} \wedge j \notin \mathcal{A}_b, j = 1 \ldots p\right\}$
    if $k^* \neq k$ **then**
      Add new branch
        $B = B + 1$
      Complete LARS step for B^{th} branch with $\mathcal{A}_B = \{\mathcal{A}, k^*\}$ omitting observations $\mathcal{O}_B$
      Check for collinearity in $\mathcal{A}_B$
Rank cases
Rank variables
Nominate cases and variables

vidual observations affect the variable selection process. The LARS algorithm's sequential inclusion of variables provides a means by which each case's influence can be tested against each intermediary step within the process. It is possible to identify the effect of a small set of observations on each decision, rather than looking at how a change in the set of observations affects the entire path. Although it may be useful to compare entire Lasso paths, it is much more difficult to tease out significant and insignificant changes because the whole path provides the accummulated affect of thousands of decisions. Comparisons are also easily understandable and interpretable for analysts as each one focuses on a binary set of choices, e.g. with these observations LARS chose variable A as most important in step k, but without them LARS chose variable B.

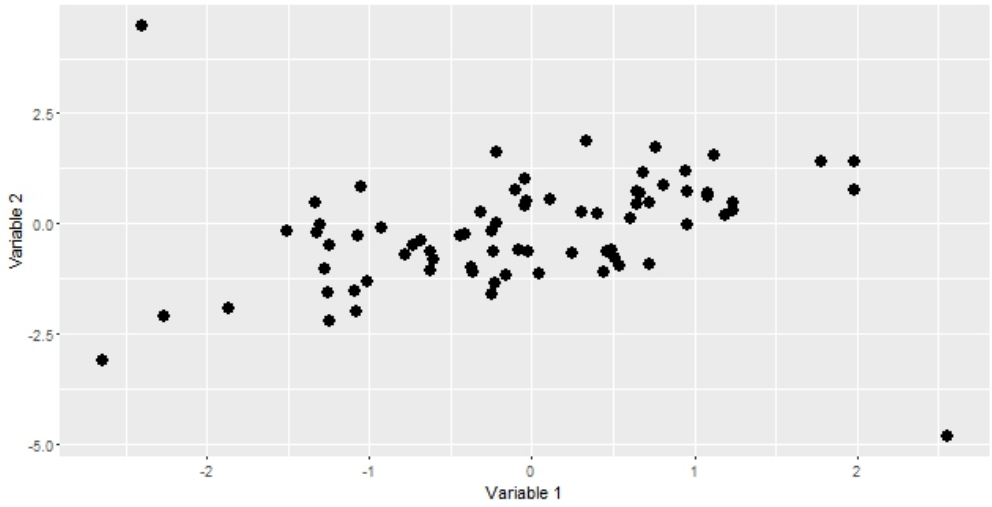

Figure 3.1: Visual representation of underlying collinearity maseked by the presence of unusual observations in the low eigenvector direction.

The LARS algorithm has also been shown to be stable with multicollinearity among the predictors (see [33]). This is advantageous for two reasons: 1) many data sets requiring sparse variable selection have at least some collinear subsets of predictors; and 2) it is common to induce or uncover latent collinearity through the removal of observations. The second issue can be easily visualized

in Figure 3.1. In this image, it is clear that there is a positive underlying correlation between Variables 1 and 2, and that there are two observations that do not seem to fit this trend. With those two observations included, the calculated correlation gives no indication of collinearity ($r = 0.24$), however if those two observations are removed the two variables appear quite correlated ($r = 0.65$). The `laron` package based on this algorithm (described in Chapter 3) tracks collinearity as the algorithm progresses (recording the VIFs and condition numbers for the design matrix) and may be accounted for after the fact, but due to the increased likelihood of encountering multicollinearity among predictors it is preferable to work with a variable selection algorithm that will be stable when and if collinearity arises.

3.1.2 Theoretical Example

The general form of this process may be best seen through a partial example. Suppose we have a simple data set consisting of design matrix $\mathbf{X}$ with 100 observations and 80 variables, of which we expect few to be important for our response variable $\mathbf{y}$. We initially determine that variable 6 is most correlated with $\mathbf{y}$, so we add it to the model. We calculate Cook's distance on all observations in the linear model between $\mathbf{y}$ and X_6, and find no values to be above our cutoff of $\frac{4}{(100-1-1)}$. Therefore we proceed to the next step in the LARS process: let the current active set for the trunk $\mathcal{A}_0 = \{6\}$, update the direction for the coefficient estimate

$$\mathbf{w}_{\mathcal{A}_0} = \left[\overbrace{0, \ldots, 0}^{5}, 1, \overbrace{0, \ldots, 0}^{74}\right],$$

the scalar multiplier for the direction γ as defined in Section 2.3.2, the coefficient estimate

$$\hat{\beta}_{0,1} = \mathbf{0} + \hat{\gamma}\mathbf{w}_{\mathcal{A}_0},$$

and all corresponding fitted values and residuals $\mathbf{e}_0$. In general throughout this example, subscripts are meant to distinguish different branches from each other. In the case of the coefficient estimates which are recursively defined, the second subscript indexes the step.

Now check for the next variable to add to the model. Calculate $\mathbf{C}_0 = \mathbf{e}_0^T\mathbf{X}$ and find the new variable's index $k = \left\{j : |c_j^*| = \mathbf{C}_{\max}^* \wedge j \notin \mathcal{A}_0,\ j = 1 \dots p\right\}$; suppose we should add variable 2. We calculate Cook's distances on the model regressing $\mathbf{y}$ on X_6 and X_2 and find that the 12^{th} observation has a Cook's distance of 1.5, much larger than our cutoff of $\frac{4}{(100-2-1)} \approx 0.041$. Therefore we set aside case 12, calculate $\mathbf{C}$, and determine that without this observation, variable 1 is most correlated with the residuals $\mathbf{e}_0$. This new branch (with index 1) shares all of the details of branch 0 ("the trunk") for the first step, but diverges from it at step 2. For this new branch 1, define the outlier set $\mathcal{O}_1 = \{12\}$. This branch and all subsequent branches use only those observations in $\mathbf{X}$ and $\mathbf{y}$ that do not belong to $\mathcal{O}_1$, denoted as $\mathbf{X}^*$ and $\mathbf{y}^*$.

For both branches, determine the new active sets, coefficient directions, scalar multipliers, coefficient estimates, and residuals:

$$\begin{aligned}
\mathcal{A}_0 &= \{2, 6\} & \mathcal{A}_1 &= \{1, 6\} \\
\mathbf{w}_{\mathcal{A}_0} &= \left[0, \tfrac{1}{\sqrt{2}}, \overbrace{0, \dots, 0}^{3}, \tfrac{1}{\sqrt{2}}, \overbrace{0, \dots, 0}^{74}\right] & \mathbf{w}_{\mathcal{A}_1} &= \left[\tfrac{1}{\sqrt{2}}, \overbrace{0, \dots, 0}^{4}, \tfrac{1}{\sqrt{2}}, \overbrace{0, \dots, 0}^{74}\right] \\
\hat{\beta}_{0,2} &= \hat{\beta}_{0,1} + \hat{\gamma}_0\mathbf{w}_{\mathcal{A}_0} & \hat{\beta}_{1,2} &= \hat{\beta}_{0,1} + \hat{\gamma}_1\mathbf{w}_{\mathcal{A}_1} \\
\mathbf{e}_0 &= \mathbf{y} - \mathbf{X}^T\hat{\beta}_{0,2} & \mathbf{e}_1 &= \mathbf{y} - \mathbf{X}^T\hat{\beta}_{1,2}
\end{aligned}$$

Both branches now continue with their next LARS step. Starting with the trunk, suppose the next variable to be added is the 10^{th}. Two observations show high influence: observations 3 and 12 (which, although removed from the dataset in branch 1, remains in the model for the trunk). With these observations removed, the 1^{st} variable would be added; since this variable is different than than the one we selected with all observations included a new branch (branch 2) is added.

Now the 3^{rd} step of branch 1 (the one that broke off in step 2) must also be completed. Correlations are calculated, and variable 4 is selected as the next to be introduced into the model. After checking the Cook's distance measures for all the remaining observations (i.e. all cases except the 12^{th}), case 3 is found to have influence larger than the cutoff of $\frac{4}{99-3-1}$. When this observation is removed, however, variable 4 is still found to be most correlated with the residuals $\mathbf{e}_1$ from the previous step so no new branch is created.

This process continues until all branches are deemed to be full.

3.2 Simulation Studies

3.2.1 Simulation Set-up

For these simulations, we generate the majority of data (henceforth known as "clean" data) according to the standard underlying model with n observations

and p variables

$$\begin{aligned} \mathbf{X}_C &\sim N_p(\mathbf{0}, \Sigma) \\ \mathbf{y}_C &= \mathbf{X}\beta + \sigma\epsilon \end{aligned}$$

where $\epsilon \sim N(0, 1)$, σ is the residual standard deviation, and Σ is the covariance matrix normalized such that $Var(X_j) = 1$.

Outliers are formed by manipulating one or both of the following two case characteristics:

1. the leverage, usually measured by the value $h_i = x_i(\mathbf{X}^T\mathbf{X})^{-1}x_i^T$, i.e. the i^{th} diagonal element of the hat matrix $H = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ where $0 \leq h_i \leq 1$; and
2. the residual or error from the model $e_i = y - x_i^T\beta$.

The residual is easily manipulated, and so will be addressed first. Once there is a design matrix $\mathbf{X}_{\mathcal{O}}$, the residual can be controlled using a multiplier δ_y

$$\mathbf{y}_{\mathcal{O}} = \mathbf{X}_{\mathcal{O}}\beta + \delta_y\sigma + \eta_y\epsilon$$

where ϵ is standard normal and $\eta_y = \frac{range(\mathbf{y}_C)}{50}$ provides a small amount of jittering. As outlier clusters (multiple observations occupying the same general area of space that is unusual to the rest of the data) are generally more difficult to identify for leave-one-out diagnostics like those generally used in LARON, multiple outliers were always modeled in clusters.

To adjust the leverage component, consider the singular value decomposition (SVD) of a low-dimensional ($n > p$) predictor matrix

$$\mathbf{X} = UDV^T$$

where U is the $n \times p$ matrix of left singular vectors, V is the $p \times p$ matrix of right singular values, and D is the $p \times p$ diagonal matrix of singular values $d_1 \geq d_2 \geq$

$\ldots \geq d_p \geq 0$, with U and V idempotent. (Note that the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$ generates matrices V and D^2.) In the context of linear regression, the U matrix is directly connected to the leverage, since $H = UU^T$. Therefore the leverage values are entirely expressible using only the values of U

$$h_i = \sum_j u_{ij}^2.$$

The advantage of working with leverage in terms of the SVD is that one not only controls the amount of leverage but also the location in the predictor space (in terms of the eigenvectors) where the leverage occurs. Recall that a leverage value greater than $\frac{2p}{n}$ is generally considered suspect.

An additional level of complexity is added when the data set is high-dimensional, i.e. $n < p$. Here we utilize the results from the Eckart-Young Theorem [19], which stated that, supposing $rank(\mathbf{X}) = n$, the SVD can be coerced into a block form, replacing all singular values beyond the first n with 0:

$$\mathbf{X} = \left[\begin{array}{c|c} \underbrace{U_1}_{n\times n} & \underbrace{U_2}_{n\times(p-n)} \end{array}\right] \left[\begin{array}{c|c} \overbrace{D_1}^{n\times n} & \mathbf{0} \\ \hline \mathbf{0} & \underbrace{\mathbf{0}}_{(p-n)\times(p-n)} \end{array}\right] \left[\begin{array}{c} \overbrace{V_1^T}^{n\times p} \\ \hline \underbrace{V_2}_{(p-n)\times p} \end{array}\right]$$

Therefore we ignore all but the first n singular values and right singular vectors for possible leverage directions. This also indicates that all but the first n columns of U should be used for calculating leverage, else necessitating division by 0, and that every leverage value must be equivalently 1.

This method for creating data would give us complete control over our data except for one minor issue: the LARS routine requires that each column be normed to 1. Although this scaling will not change the leverage value of the observation, it will alter its location in predictor space. However, this effect is

not significant enough to nullify any conclusions. Because of this, generally I will focus on leverage points in a single eigenvector direction (major or minor) so outlying predictors take the form

$$\mathbf{X}_{\mathcal{O}} = \delta_x d_j \mathbf{v}_j + \eta_x \epsilon$$

where ϵ is standard normal as before and η_x is calculated similarly to η_y.

Data were sampled from a multivariate normal with mean 0 with both uncorrelated and highly correlated ($r = 0.9$) autoregressive covariance structures. Residual variance σ^2 was selected to produce an average signal-to-noise (SNR) ratio (calculated according to the [15]) of approximately 11 for all clean observations, where

$$\text{SNR} = \frac{(\bar{\mathbf{y}}\mathbf{1} - \mathbf{X}\beta)^T(\bar{\mathbf{y}}\mathbf{1} - \mathbf{X}\beta)}{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}.$$

Low dimensional sets consisted of 100 observations in 10 predictors. The coefficient vector was defined as $\beta = (3, -5, 0, 0, 2, 0, -1, 0, 0, 0)^T$ which translated to $\sigma^2 = 20^2$ for the uncorrelated data, and $\sigma^2 = 7^2$ when correlated.

3.2.2 Outlier Detection

In the case of outlier detection, each observation may fall into one of the following four classes:

True Positive (TP): Correctly identified an outlier

True Negative (TN): Correctly identified a clean observation

False Positive (FP): Incorrectly identified a clean observation as outlying

False Negative (FN): Incorrectly failed to identify an outlier

This is often summarized in what may be called a "confusion" matrix, given in Figure 3.2.

	Outlier	Clean Point
Nominated	TP	FP
Not Nominated	FN	TN

Figure 3.2: Confusion Matrix

The receiver operating characteristic (ROC) curve is a common way to visualize the detection capabilities of detection algorithms. It plots the false alarm rate (FAR) against the detection rate (DR) achieved by a method, where

$$\begin{aligned} \text{FAR} &= \frac{\text{FP}}{\text{FP} + \text{TN}} \\ &= \frac{\text{FP}}{\#\left\{\text{clean points}\right\}} \end{aligned}$$

and

$$\begin{aligned} \text{DR} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ &= \frac{\text{TP}}{\#\left\{\text{outliers}\right\}}. \end{aligned}$$

A perfect ROC curve follows the piece-wise linear curve tracing out the left and top boundaries of the unit square. Curves close to the line $y = x$ indicate an indifferent nominator (essentially no difference from random guessing), while those closer to the bottom right corner indicate a highly faulty nominator where outliers are less likely to be nominated than clean observations. For a more detailed explanation of this topic, see [25].

The ROC curves in in Figures 3.3 through 3.5 compare the LARON algorithm to post-hoc outlier nominators based on CV-selected fits using the LARS and CD

algorithms described in Sections 2.1 and 2.3. Although the 1SE rule discussed earlier is generally preferable and almost exclusively used in conjunction with CV-selection due to increased stability and sparsity properties, the penalty parameter generating the minimum error was also used for comparison purposes. The latter is more likely to be useful in outlier detection specifically because it tends to be less sparse, and it is advantageous to examine as large of a subspace of the predictors as possible to increase detection probabilities. Detection capabilities were averaged over simulations conducted in both low and high dimensional data sets as outlined in Section 3.2.1.

To generate the ROC curves, the contamination level was set to be approximately 5%, with five outliers in the low dimensional data set and three in the high dimensional. Outliers were created using values of $\delta_x = 1.2$ and $\delta_y = 4$ along both major and minor eigenvector directions. 100 data sets were sampled, and false alarm and detection rates were calculated based on influence rankings using Cook's distance. If CV-selection induced a null model such that no influence rankings could be calculated, outliers were given rankings spread uniformly through the set of observations. These results may be seen in Figures 3.3 and 3.4.

The clear stepwise pattern, generally visible in the Lasso curves, is due to the artificial uniform distribution of outliers when the null model is selected. Models which tend to over-penalize to the null model will have a more pronounced step-wise pattern. In the bottom right of Figure 3.3, Lasso fits using the 1SE model never selected a single variable for influential outliers in a minor eigenvector direction for highly correlated predictors.

In low dimensions, the LARON method clearly outperforms post-hoc diag-

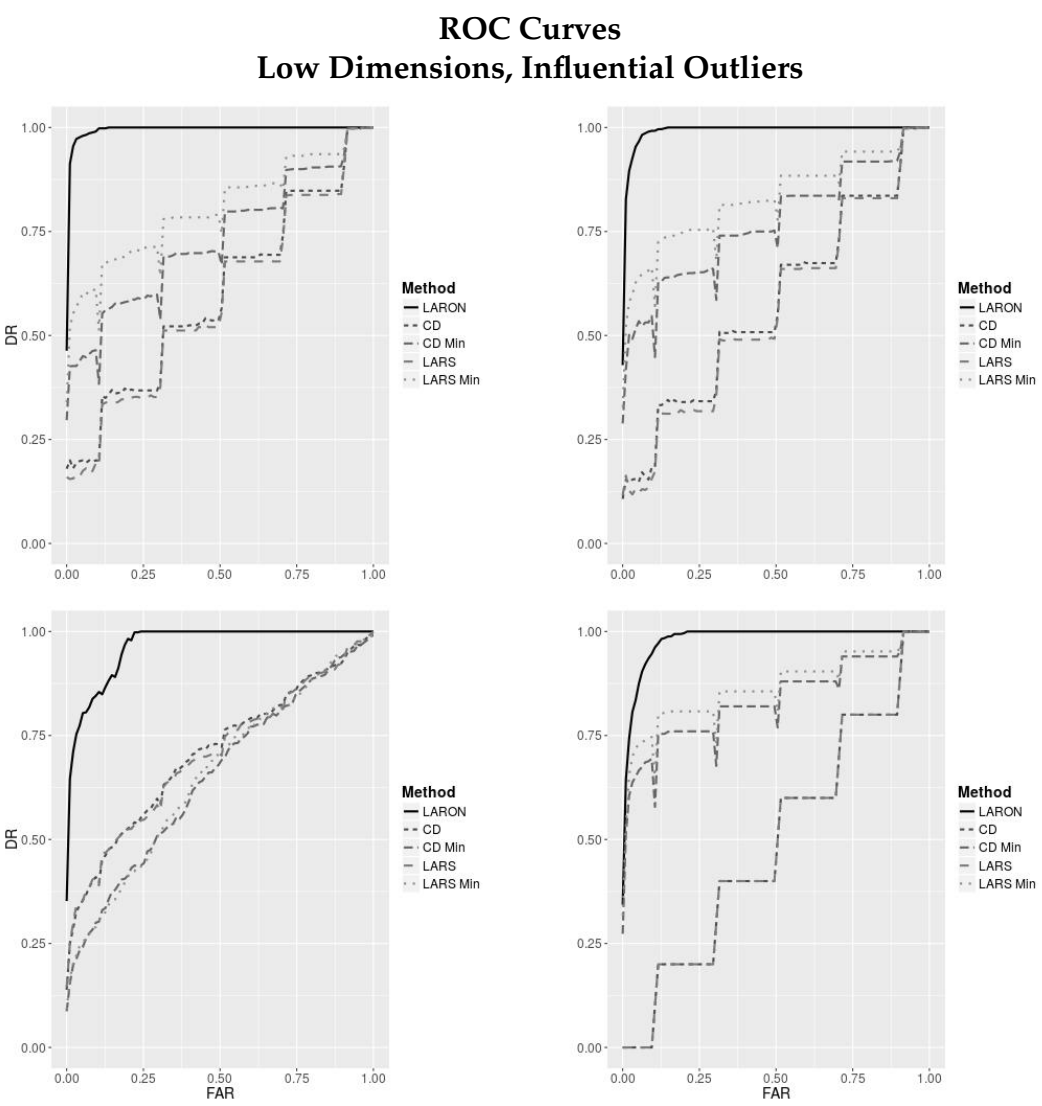

Figure 3.3: ROC curves comparing outlier detection procedures in low-dimensional data using LARON and post-hoc analyses of LARS and CD CV-selected fits. In the top row, covariates are uncorrelated; the bottom row has an autoregressive correlation structure with $r = 0.9$. Outliers in the left column were moved along the major eigenvector direction; in the right column, along the minor direction.

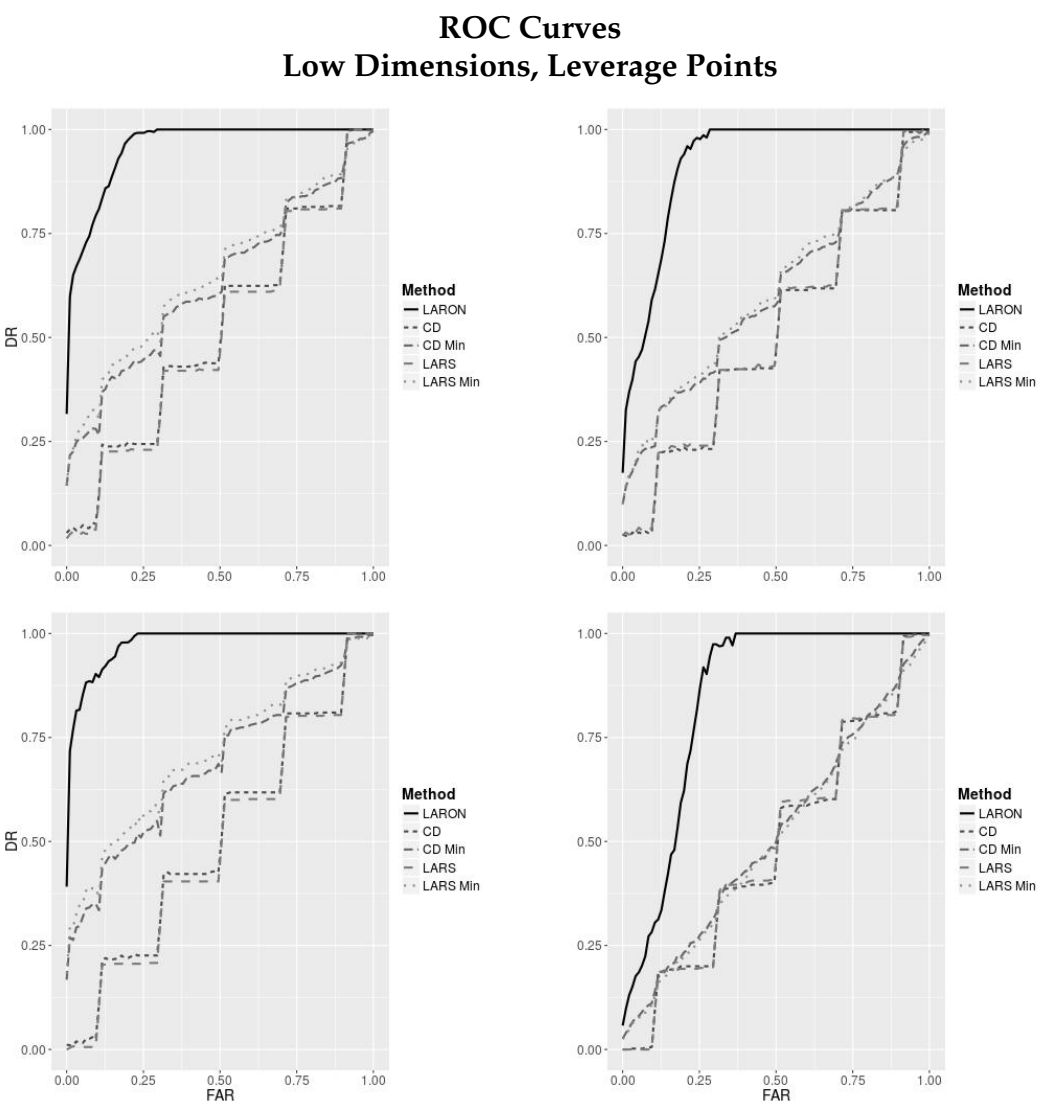

Figure 3.4: ROC curves comparing outlier detection procedures in low-dimensional data using LARON and post-hoc analyses of LARS and CD CV-selected fits. In the top row, covariates are uncorrelated; the bottom row has an autoregressive correlation structure with $r = 0.9$. Outliers in the left column were moved along the major eigenvector direction; in the right column, along the minor direction.

nostics on a Lasso fit for both influential and leverage outliers. The object is to reach the maximum detection rate at the earliest possible false alarm rate. LARON tended to detect all five outliers by the time the FAR reached about 12% for uncorrelated data and before 25% in correlated; the post-hoc analyses generally had false alarm rates of 88% at best before all 5 outliers were detected. Leverage outliers were more difficult to detect than influential points. LARON tended to detect all outliers when the false alarm rate was between 25% and 30%. The 1SE rule algorithms almost invariably selected the null model; the minimum rule did not suffer from that issue as much, however its outlier detection performance was not improved as a result.

All methods are generally better able to detect outliers in major eigenvector directions rather than minor, and in uncorrelated data rather than highly correlated. The one exception was that the minimum rule tended to perform slightly better for correlated data in the minor eigenvector direction. It is similarly clear that the 1SE rule (denoted simply as LARS or CD) is significantly worse in terms of outlier detection than using the minimum penalty parameter. It also appears as though the LARS algorithm has a very slight advantage over CD in post-hoc analysis.

High dimensional data were sampled to have size $n = 50$ and $p = 100$. When in these high dimensions, using the maximum leverage value is unstable, as all observations exhibit "high leverage". When $n \leq p$, the leverage statistic $h_i \equiv 1$ for every observation. In such instances, influence measures such as Cook's distance and DFFITS involve division by $(1 - h_i)$. Cook's distance is also known to follow an F distribution with degrees of freedom $d_1 = p$ and $d_2 = n - p$ under the assumption that the error values ϵ are normally distributed. In order to maintain

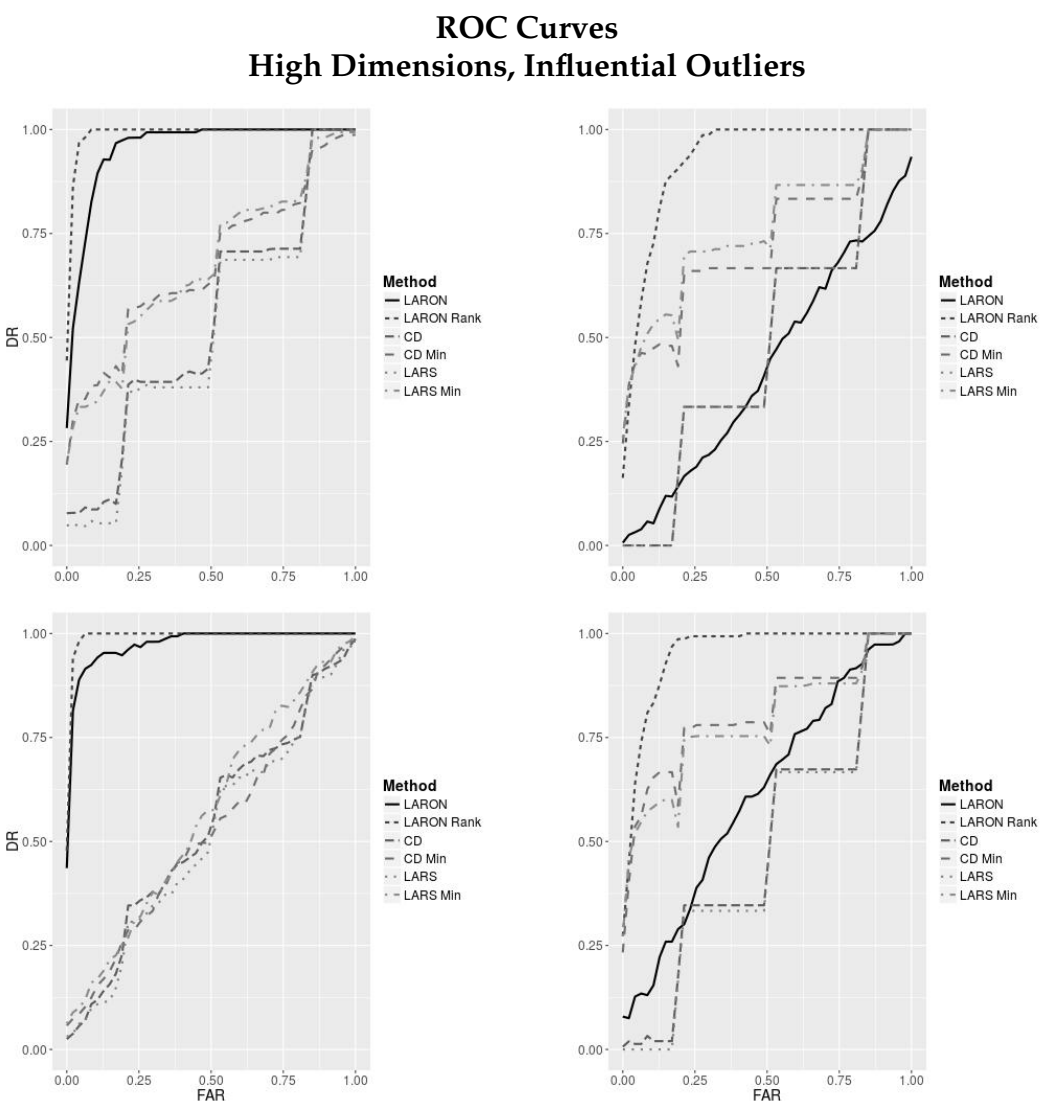

Figure 3.5: ROC curves comparing outlier detection procedures in high-dimensional data using LARON and post-hoc analyses of LARS and CD CV-selected fits. In the top row, covariates are uncorrelated; the bottom row has an autoregressive correlation structure with $r = 0.9$. Outliers in the left column were moved along the major eigenvector direction; in the right column, along the minor direction.

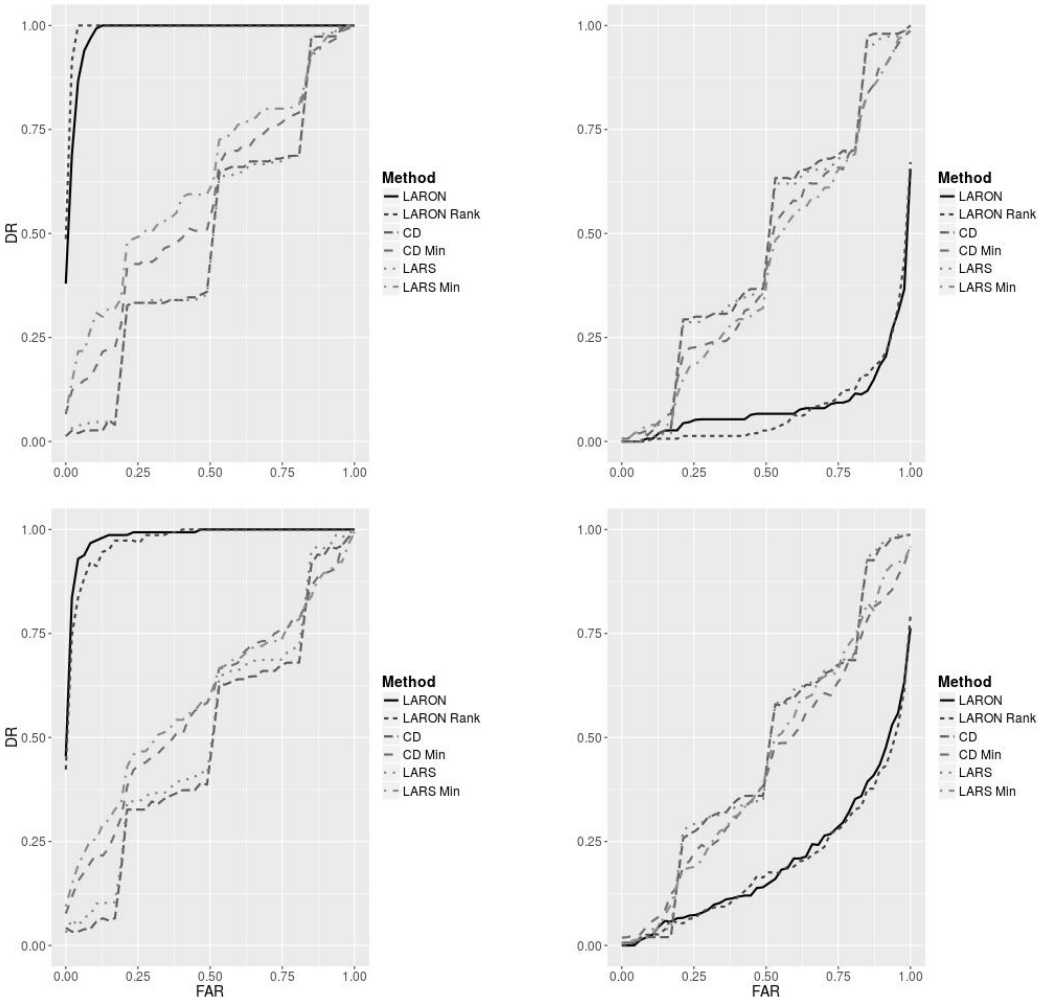

Figure 3.6: ROC curves comparing outlier detection procedures in high-dimensional data using LARON and post-hoc analyses of LARS and CD CV-selected fits. In the top row, covariates are uncorrelated; the bottom row has an autoregressive correlation structure with $r = 0.9$. Outliers in the left column were moved along the major eigenvector direction; in the right column, along the minor direction.

finite variance, therefore, influence measures are only measured if the restriction $n - p > 4$ is satisfied. Although this does maintain distributional consistency, all observations tend to have inflated influence measures as the available degrees of freedom come close to 4. To account for this, a ranking statistic is utilized incorporating the selection ratio of the observation. This selection ratio SR is defined to be

$$SR_i = \frac{\text{\# of branches omitting observation } i}{\text{\# of branches}}.$$

As can be seen in the following ROC curves, this ranking does significantly better than all other methods, even though the influence measure alone does worse even than detection using post-hoc diagnostics with the minimum penalty parameter. This improvement is not as noticeable when the outlier is shifted in the major eigenvector direction, however the improvement is quite significant when the outlier is in the minor eigenvector direction.

In high dimensions, LARON performs well using either the ranking value or the influence value for detecting any outliers in a major eigenvector direction. For influential outliers, the LARON influence value alone performs essentially indistinguishably from the arbitrary nomination induced by the null model selection of the 1SE rule. The LARON ranking statistic outperforms all other methods in this scenario.

The most interesting result was the poor performance detecting high leverage outliers in a minor eigenvalue direction. Non-LARON methods perform essentially arbitrary nomination. The LARON methods, however, perform worse than simple random guessing; the high leverage points are less likely to be nominated than non-outlier points. This indicates that the variable selection in the Lasso does not tend to bend itself towards strictly leverage points. This was

not as noticeable in low dimensions because the entire predictor space will be examined eventually; i.e. in high dimensions, the variables with the largest components in the minor eigenvector direction are the least likely to be selected in the model.

3.3 Examples

3.3.1 Simulated Example

A low dimensional simulated data set provides a useful demonstration of the LARON process. This data set was simulated with $n = 100$ observations and $p = 12$ predictors. The first four predictors were selected to be the true active set, with coefficients of 5 and alternating signs. The residual error σ was set to 1 and the predictors were sampled from an uncorrelated multivariate normal distribution with identity covariance. After building the data set according to Section 3.2.1, one observation (case 91) was arbitrarily selected to be an influential point. It was moved along a direction dominant in variables 5, 6, and 7 three times the distance of the maximum observation value in that direction. The residual was augmented by $\delta_y = 5$.

Figure 3.7 plots the change in Residual Sum of Squares as new variables are added into the model. The black "trunk" of this tree represents the path traced by the full data set; each colored "branch" indicates that a set of outliers has been removed from the model according to the rule proposed. For instance, the first branch indicated by the red line picked up in step 3 that observation 91 had a large Cook's distance and, if it were ignored, would choose to include variable

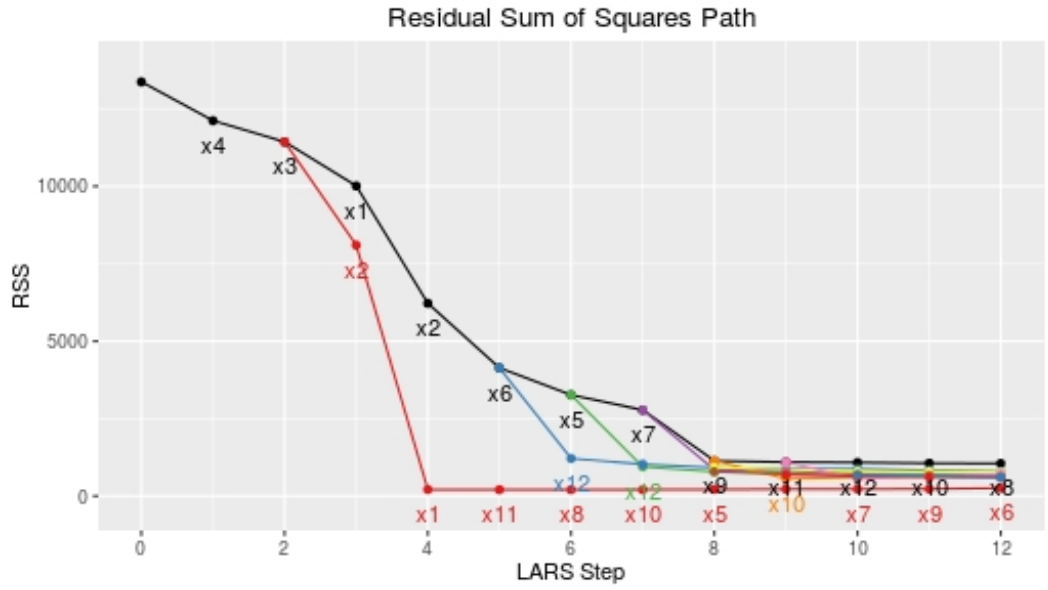

Figure 3.7: LARON RSS path for simulated data set

2 rather than variable 1. There is a small reduction in the average prediction error at that point, but the significance of dropping that outlier isn't really seen until the next step. Both models include all four relevant variables by the forth step, but the branch omitting the outlier drops to an average prediction error unreachable by the full data set. The branch also stabilizes at this point such that additional variables add little to the model. The full model, on the other hand, doesn't stabilize until four extraneous variables have been added; notably, the three predictors dominant in the outlying direction of case 91 plus one other. The influence of the outlier on the variable selection in this case is obvious.

This is not to be viewed as a traditional lasso path, which is generally indexed by the penalty parameter in the Lagrangian or fractional form, nor does it show the relevant coefficient estimates at each stage. The purpose of this plot is not to visualize the full model itself, but rather to observe the relative impact on prediction and variable selection that arise from the omission of observations from consideration. The reasoning behind choosing the LARS step as the index

Table 3.1: Description of variables in the diabetes dataset.

Variable ID	Code	Description
1	age	Age in years
2	sex	Gender (1=male; 2=female)
3	bmi	Body mass index
4	map	Average blood pressure
5	tc	Blood serum 1
6	ldl	Blood serum 2: Blood low density lipid (LDL) level
7	hdl	Blood serum 3: Blood high density lipid (HDL) level
8	tch	Blood serum 4
9	ltg	Blood serum 5
10	glu	Blood serum 6: Blood glucose level
Response	y	Disease progression 1 year after baseline

comes from its easy visibility in terms of variable selection and active set size.

When examining the coefficients produced by the paths in figure 3.7, the first branch (in red) shows essentially no change in coefficients after step 4. Although additional variables are nominally entered into the model, their coefficient estimates are nearly 0. The 4^{th} step in fact has an s value of about 0.98, and so would be located in the far right of a plot indexed by s. The trunk, however, continues to see significant changes to its β_k until step 8; the 4^{th} step would in fact be referenced at $s \approx 0.44$, located to the left of center in a plot referenced by s. It becomes confusing to compare branches on this scale since variable selections at the same step could be separated by a large portion of the graph, branches have drastically different densities of nodes in the main part of the graph and the reference β_{full} may differ.

3.3.2 Diabetes Data

The diabetes dataset was used in the seminal LARS paper [20]. This dataset consists of 10 predictor variables and one response variable, described in table 3.1, measured on 442 diabetes patients. In their analysis, they ended up choosing a model with 7 variables when selecting based on minimizing Mallow's C_p.

It is important to note that there are several areas of concern within this dataset. Firstly, the response variable should be log-transformed to diminish heteroscedasticitiy and curvature in the residuals. Secondly, using standard linear modeling methodology, this dataset suffers from extensive multicollinearity; a known issue with lasso modeling.

When forming the nominating tree shown in figure 3.8, it strikes one at first glance that it is much simpler than the one constructed by simulation. It is also striking that in all three cases where outliers are nominated and removed, the model attempts to select variable 6 (ldl) whereas it would not be selected un-

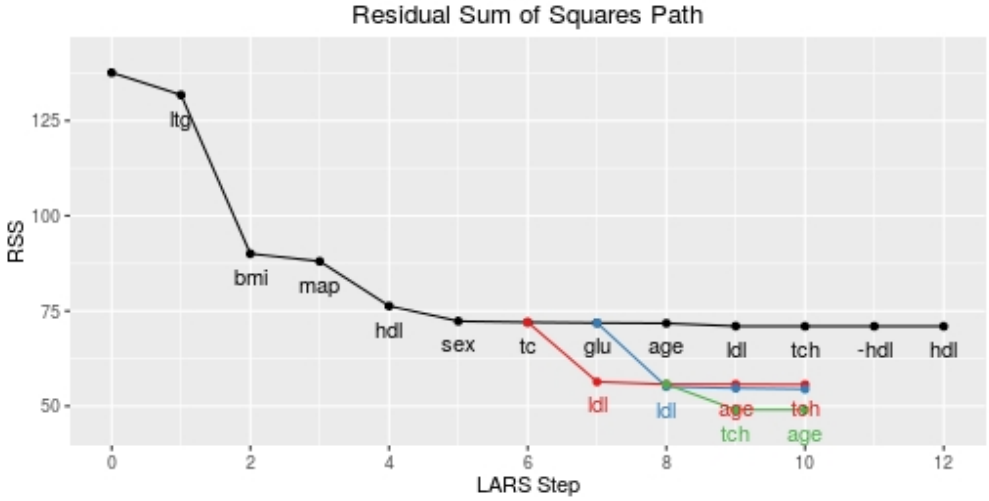

Figure 3.8: Outlier nomination RSS for diabetes dataset with log-transformed response.

til step 9 under normal circumstances (and, as it turns out, eventually dropped from the model since the CV-selected model included only the first 7 variables). When performing an analysis, it certainly makes one wonder why this variable would not ordinarily be included, but small fluctuations in the set of observations makes it a much more desirable predictor.

The variance inflation factors for each predictor are outlined in table 3.2, along with the variable with the largest pairwise correlation. If more than one variable had a correlation greater than 0.6, it was also included in the table. Note that the variable sex was omitted, as it is not a quantitative measure.

Table 3.2: Variance inflation factors and largest pairwise correlations among the diabetes predictors. In cases where there were multiple correlations above 0.6, all were listed.

Code	VIF	Highest Pairwise Correlations
age	1.22	map *(0.34)*
bmi	1.51	ltg *(0.45)*
map	1.46	bmi *(0.40)*
tc	59.20	ldl *(0.90)*
ldl	39.19	tc *(0.90)*; tch *(0.66)*
hdl	15.40	tch *(-0.74)*
tch	8.89	hdl *(-0.74)*; ldl *(0.66)*; ltg *(0.62)*
ltg	10.08	tch *(0.62)*
glu	1.48	ltg *(0.46)*

In the table, as many as half of the predictors exhibit signs of collinearity. Most notably, the tc and ldl variables are extremely pairwise collinear with each other. When tc is omitted, all vifs shrink below 5 excpet tch which maintains a relatively high VIF of 7.82. Since this is also strongly correlated with hdl, ldl, and ltg, it's unlikely to be contributing much unique information, so it would probably be advisable to drop this one as well. Once both tc and tch have been removed from the model, LARON no longer detects any outlying cases and ldl improves in importance, although the first five variables to enter the model (ltg,

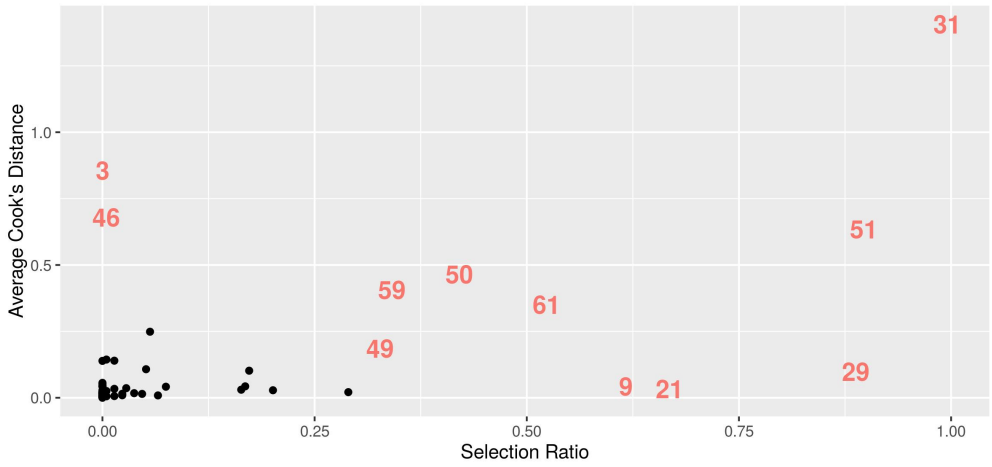

Figure 3.9: Selection ratio versus average Cook's distance from the LARON analysis of the riboflavin data. See Table 3.3 for case names.

bmi, map, hdl, and sex) still maintain their relative importance.

3.3.3 Riboflavin Data

Several recent papers have analyzed a data set provided by DSM (Switzerland) and made publicly available in R by Bühlmann et al. [9]. A sample of 71 specimens of Bacillus subtilis "that were hybridized repeatedly during a fed-batch fermentation process in which different engineered strains and strains grown under different fermentation conditions" [9]. The response variable is the log-transformed riboflavin production rate and predictors consist of 4,088 gene expression levels, also on a log scale. This data set provides an interesting case study with which to test the LARON process as it has several observations that can be visually separated from the rest of the data (see, for example, Figure 3.10 from Bar, Booth, and Wells [4]) as well as heavy collinearity among some of the predictors.

This data set has already been analyzed using various techniques for vari-

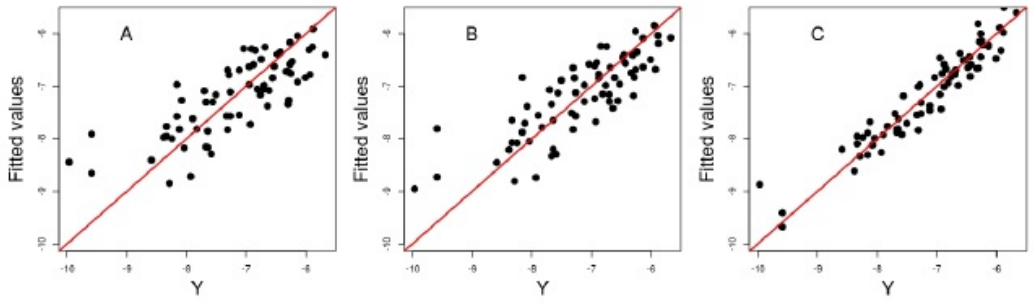

Figure 3.10: Fitted v. true response values of models selected for the riboflavin data by A – Bühlmann et al. [9], B – Hebiri and Lederer [33] , and C – Bar et al. [4]. Originally printed in [4].

able selection in high dimensional data sets. Bühlmann et al. performed multiple analyses on the data, including a Lasso stability selection procedure which determines the most commonly selected variables in multiple Lasso fits. Hebiri and Lederer [33] compared a standard CV-selected Lasso model with their TREX[1] model which is self-regulating without the need of selecting a tuning parameter or placing extra restrictions on the nature of the design matrix. Bar, Booth, and Wells [4] implement an empirical Bayes (EB) procedure with a latent diagonal matrix of coefficient signs. The analyses of [9] and [4] perfomed relatively sparse variable selection (3 and 7 predictors respectively) while [33] used a fixed active set size of 20 for comparison purposes. I also fit the full data using standard Lasso estimation with the 1SE rule for 10-fold CV-selected penalty parameters. The resulting model included 31 variables, many of which had been previously selected by the other models discussed. Full model fits may be found in Table B.7, and models accounting for correlations between predictors in 3.6.

To compare model fits, the response values were plotted against the fitted values in Figures 3.10 and 3.11. An ideal model would show points falling

[1]I must commend Hebiri and Lederer for having manipulated one of the longest names for a method with a most concise and attractive acronym. The name TREX is derived from "**T**uning-free **R**egression that adapts to the **E**ntire noise and the design matrix **X**.

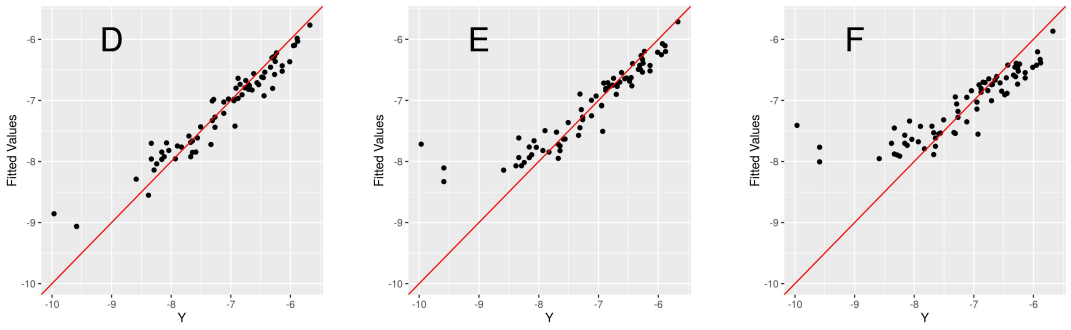

Figure 3.11: Fitted v. true response values of models selected for the riboflavin data with CV-selected penalty parameters using the 1SE rule. Plot D uses all observations, plot E omitted cases 29, 31, and 51, and plot F omitted 59 in addition to the other three.

perfectly along the line $y = x$. All plots have been placed on the same scale for reference. In all six plots it is possible to see three outliers: cases 29 (knh_661_BS5009_Fbat1374_PT48_1.Repl._No.4_rep_cDNA_new Yeast.CEL), 31 (knh_663_BS5009_Fbat1379_PT48_2.Repl._No.8_rep_cDNA_new Yeast.CEL), and 51 (knhb_091_Fbat284PT48.CEL). Given the extensive names given to these subjects, I will generally refer to them by case number or by using the first two elements of the name separated by underscores (e.g. knh_661). Full case names may be found in Table B.8. For simplicity, models which include all observations will be termed a full Lasso fit, those omitting cases 29, 31, and 51 the LARON—3, and LARON—4 when additionally dropping case 61.

These data were analyzed with the LARON algorithm, and outliers were nominated using several nomination measures and probability cutoffs (see Table 3.3) The selection ratio perfectly found the three outliers visible by eye in Figures 3.10 and 3.11, even under the stricter Chebychev probability cutoff. When those points are set aside, the model fit changes enough to consider these observations to be outliers. The plot of average Cook's distance by selection ratio (Figure 3.9) indicates that other subjects are worth investigating. Observations

Table 3.3: Outlier nomination under various nomination criteria. Highlighted observations are those which produced notable changes in the model when removed.

		Chebychev			Normal		
ID	Name	Max D	Avg D	SR	Max D	Avg D	SR
3	b_Fbat...48		•		•	•	
9	knh_492...						•
21	knh_508...						•
23	knh_512...				•		
29	knh_663...		•	•		•	•
31	knh_661...			•			•
41	knh_688...				•		
43	knhb_049...				•		
46	knhb_046...	•			•	•	
48	knhb_049...				•		
50	knhb_090...					•	•
51	knhb_091...			•		•	•
59	knhb_181...	•			•	•	
61	knhb_187...				•		•

9 and 21 were selected frequently for their propensity to change the variable selection process, however these did not appear to significantly affect the model fit. The removal of case 61 did notably alter the trajectory of the model. Results from the LARON outlier nomination process under different nominating conditions may be found in Table 3.3.

As may be expected, the resulting nominations from the LARON algorithm are not as consistent across methods as in lower-dimensional settings. Specifically, there is less agreement between the nominations identified by selection rate, and maximum or average influence measures. In these higher-dimensional settings, the maximum influence measure is more susceptible to instability produced by the curse of the dimensionality than the others, and is therefore less likely to be a good measure for outlier nomination.

The impact of these observations on the sparsity of the selected model can

be observed in the CV-selection process showed in Figure 3.12. The top image shows the full model; in the bottom image, all four cases have been set aside. In these images, it is important to note that the value of the penalty parameter is not comparable between data sets. For example, the value of the penalty is larger for the full set of data than that omitting the first three cases, and yet selects a bigger model. From these plots, it is also evident that omitting cases 29, 31, and 51 does not result in a large change to the selected sparsity of the model despite their impact on which variables are selected for inclusion. Case 61, although yielding less impact on the variable selection (as evidenced by having a much smaller selection ratio) evinces a large impact on the sparsity selection of the model. The removal of the first three points lead to the selection of a model that is reduced in size by only four predictors (27 vs. 31 predictors); the removal of all four points, on the other hand, cuts the size of the active set by more than half, to 14 predictors.

Within the collection of selected models, there are a several variables that exhibit moderately high pairwise correlations ($r > 0.7$). Since the Lasso tends to arbitrarily select one of a correlated set, these variables have been collected into groups (with Group 1 exhibiting the highest correlation) in order to more easily compare variable selection among models. These groups are given explicitly in Table 3.4; Figure 3.14 provides the group correlation heat map, where each row is a correlated group and dots indicate membership in the group.

The coefficient values with aggregate group effects are given in Table 3.6. Several variables are common to all fitted models. Group 2 is consistently important across all models. Group 1 is generally quite important, although it is not included in the TREX model. Group 7 is only missing from the Empirical

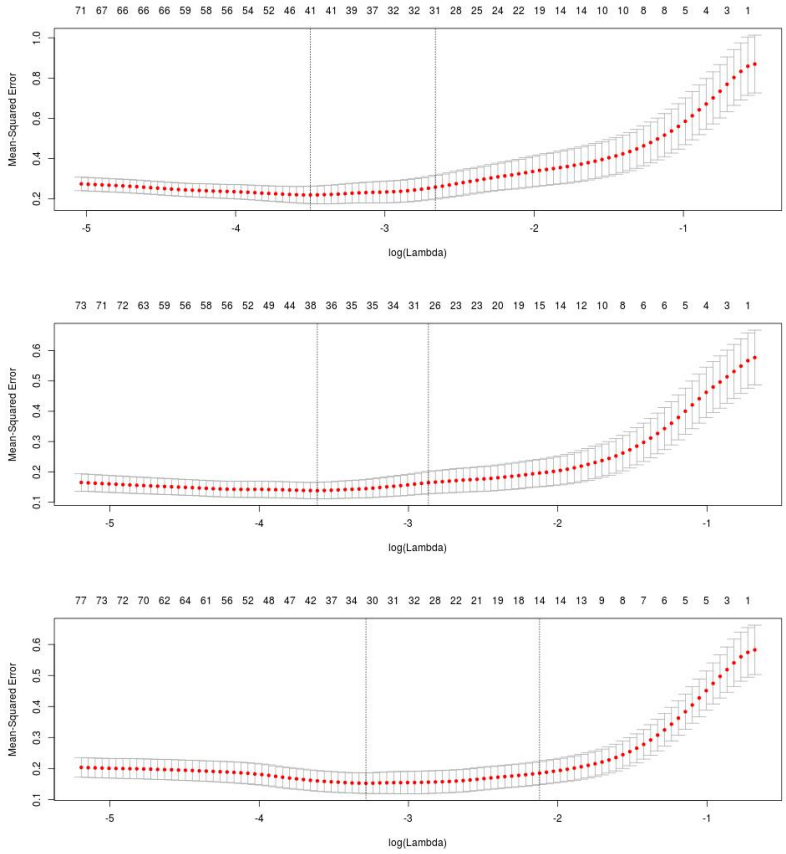

Figure 3.12: 10-fold cross-validation for (from top to bottom) the full Lasso model, LARON—3, and LARON—4. Dotted lines to the left indicate the minimum CV error penalty selection rule; those on the right indicate the 1SE rule.

Bayes model, and, like Group 1, has a relatively large coefficient value when it is included.

There are several distinct changes to the variable selection with the omission of outliers. YOAB_at, consistently one of the most important variables in models without the outliers, dropped significantly in importance. The most important predictor became instead SPOIISA_at, which was only selected for the full Lasso, but with small effect. Group 6 and Group 9 (included in both the full Lasso and the TREX models) were not selected for either LARON—3 or LARON—4; however, Group 5 entered into the model and was considered rel-

Table 3.4: Groups of correlated variables ($r > 0.7$) in the selected models for the riboflavin data.

Group ID	Variables
Group 1	XHLB_at, XTRA_at, XKDS_at, XKDN_at
Group 2	YXLD_at, YXLE_at
Group 3	ARGF_at, ARGJ_at
Group 4	YCGO_at, YCGN_at
Group 5	YTGB_at, YDAR_at
Group 6	YCDH_at, YHZA_at
Group 7	YURQ_at, YCKE_at, YJCJ_at
Group 8	YDDK_at, YDDM_at
Group 9	PKSA_at, GAPB_at

atively important for both LARON—3 and LARON—4. Group 3 (included in all three full models) dropped in importance for LARON—3 and was ignored completely in LARON—4. The YORB_i_at and IOLE_at genes were selected for both of these models with moderate coefficients, though was not selected by any full model.

There were some interesting similarities among the full Lasso, TREX, and LARON—3 models. YEBC_at was considered fairly important for the full Lasso, TREX, and LARON—3 models, but was not included for EB or LARON—4 models. The YFHE_r_at gene also had a similar effect size for all three models. However, the TREX and full Lasso model were appear to be the most similar of the five. Both included variables such as LYSC_at, YHDS_r_at, YDDH_at, YYDA_at, SPOVAA_at, and YEZB_at with fairly similar coefficient values.

In the plots of fitted values against responses in Figures 3.10 and 3.11, it is clear that athe only model which strictly follows the three noticeable cases (29, 31, and 51) is the empirical Bayes approach of [4]; however, the full Lasso model also comes close to following them. All remaining models produce significantly more moderate predicted values. All three Lasso models yield smaller variance

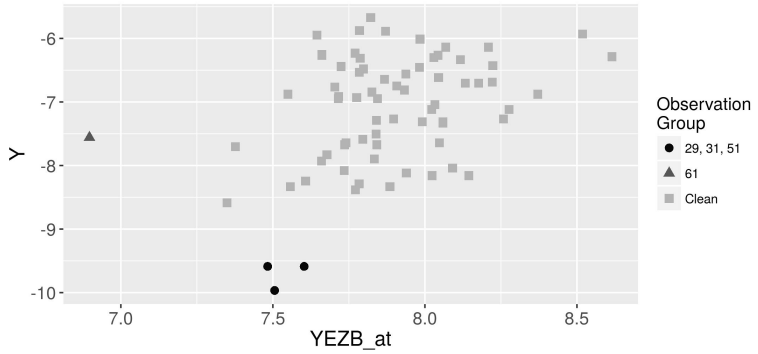

Figure 3.13: Riboflavin production v. YEZB_at predictor, the univariate direction wherein the leverage of case 61 may most clearly be seen.

around the $y = x$ line than the other three models. The models fit to data sets after some cases had been set aside tend to deviate more noticeably from those observations. Case 61 is not particularly visible in these plots as it has a very moderate response value, with $y = -7.56$. It derives its importance from its unusual behavior in the predictor space. It is most noticeably unusual in the variable YEZB_at (seen in Figure 3.13), although there is also a reasonable amount of leverage in the IOLE_at predictor as well. With the removal of these observations, YEZB_at becomes less correlated with the response (from $r = 0.394$ to 0.298) while IOLE becomes more correlated (from $r = -0.408$ to -0.483).

Table 3.5: AIC values for models fit to the riboflavin data.

AIC Test Set	Full Lasso	TREX	EB	LARON—3	LARON—4
All Observations	0.9014228	1.155693	3.404448	0.7892631	0.5376125
$-\{29, 31, 51\}$	0.9411870	1.481993	4.112980	0.8237278	0.5599170
$-\{29, 31, 51, 59\}$	0.9552343	1.512509	4.115392	0.8360186	0.5681596

Another useful tool for measuring goodness of fit is the AIC. Although there is some debate about the best mode of calculating AIC for the Lasso, I use the value as described in [72]. This equation was developed through the use of Stein's unbiased risk estimation (SURE) to find an effective degree of freedom

estimate for the Lasso.

$$AIC(\hat{\beta}) = \frac{\|\mathbf{y} - \mathbf{X}\hat{\beta}\|_2^2}{n\sigma^2} + \frac{2}{n}\left|\mathcal{A}_{\hat{\beta}}\right|$$

where σ^2 is the variance of $\mathbf{y}$ and $\left|\mathcal{A}_{\hat{\beta}}\right|$, the number of non-zero coefficient values in $\hat{\beta}$, is the estimated degrees of freedom. Results for the models have been provided in Table 3.5. It is important to note that AIC values should not be compared across different data sets (i.e. do not compare numbers down columns, but only across rows), and that a smaller AIC value is considered to be better.

It is clear from the table that the standard lasso models tend to perform better than TREX or EB for this data set, although they generally suffer from less sparse models. When tested against using all three sets of training cases, the LARON—4 consistently has the lowest AIC out of the five models. It clearly appears to have a generally tighter set of residuals compared to TREX and EB like the full Lasso and the LARON—3 model, although it is not as accurate. However, this lack of accuracy is offset by its significant decrease in model size, having 6 predictors fewer even than the TREX model.

Determining the "best" model is not trivial in this case, and it largely depends on the goal of the analysis and the view of the analyst concerning the outlying cases 29, 31, 51, and 61. It is clear that they alter both the prediction, the variable selection, and sparsity of the selected model. If these observations provide a glimpse of a small but important subgroup of Bacillus subtilis, then it appears that EB has the most success in predicting that subgroup, while the full Lasso does the best job of balancing predicting the outliers and the main group of points. If they are members of an alternative population or have values in error, than likely either LARON—3 or LARON—4 would be preferable. Models which are more sparse tend to be preferable for interpretation, so LARON—4

would have the advantage in this context, however it underperforms LARON—3 in prediction. Given the many trade-offs inherent to data analysis such as this, it is evident that one cannot depend upon "black box" analytical tools but must instead interact thoughtfully with the information available from methods such as LARON.

Table 3.6: Coefficient values for models fit to the riboflavin data, after accounting for correlated groups of predictors. Variables listed in order according to coefficient magnitude (largest to smallest). All Lasso models' penalty parameters were selected according to the 1SE rule.

Full Lasso		TREX		EB		LARON—3		LARON—4	
Intercept	1.155	*Intercept*	-7.159	*Intercept*	-7.600	*Intercept*	-5.202	*Intercept*	-8.708
YOAB_at	-0.697	*Group 2*	-0.238	YOAB_at	-1.130	SPOIISA_at	0.322	SPOIISA_at	0.370
YEBC_at	-0.428	YOAB_at	-0.168	*Group 2*	-0.920	YEBC_at	-0.289	*Group 7*	0.222
LYSC_at	-0.285	*Group 3*	-0.112	*Group 3*	-0.860	*Group 7*	0.230	*Group 2*	-0.193
Group 2	-0.282	YEBC_at	-0.088	*Group 1*	0.610	*Group 2*	-0.213	*Group 1*	0.190
Group 1	0.248	*Group 7*	0.069	YHDZ_at	0.460	*Group 1*	0.189	*Group 5*	-0.152
Group 7	0.187	*Group 4*	-0.065			*Group 8*	-0.166	IOLE_at	-0.109
YBFI_at	0.178	YEZB_at	0.049			THIC_at	-0.164	*Group 4*	-0.083
Group 8	-0.158	*Group 6*	-0.045			*Group 5*	-0.125	YORB_i_at	0.039
YCLB_at	0.157	YFHE_r_at	0.041			*Group 4*	-0.111	YHFU_at	-0.037
DNAJ_at	-0.148	LYSC_at	-0.024			YFHE_r_at	0.098	YOAB_at	-0.022
Group 3	-0.145	*Group 8*	-0.024			YPGA_at	-0.091		
YFHE_r_at	0.104	RPLL_at	-0.022			YORB_i_at	0.079		
SPOVAA_at	0.099	YYDA_at	-0.015			IOLE_at	-0.072		
YEZB_at	0.079	YBFI_at	0.007			YHAI_at	0.058		
Group 9	0.075	YHDS_r_at	0.006			CTAA_at	0.052		
YYDA_at	-0.065	*Group 9*	0.003			YOAB_at	-0.039		
YHDS_r_at	0.056	SPOVAA_at	0.003			ccpB_at	-0.026		
YQJU_at	0.037	YDDH_at	-0.001			*Group 3*	-0.024		
YCLF_at	-0.034					YHCB_at	-0.024		
YFIR_at	0.034					YCLB_at	0.021		
YRVJ_at	-0.029					YKBA_at	0.020		
YKBA_at	0.028					YVFK_at	0.011		
SPOIISA_at	0.023								
Group 4	-0.015								
YDDH_at	-0.006								
Group 6	-0.000								

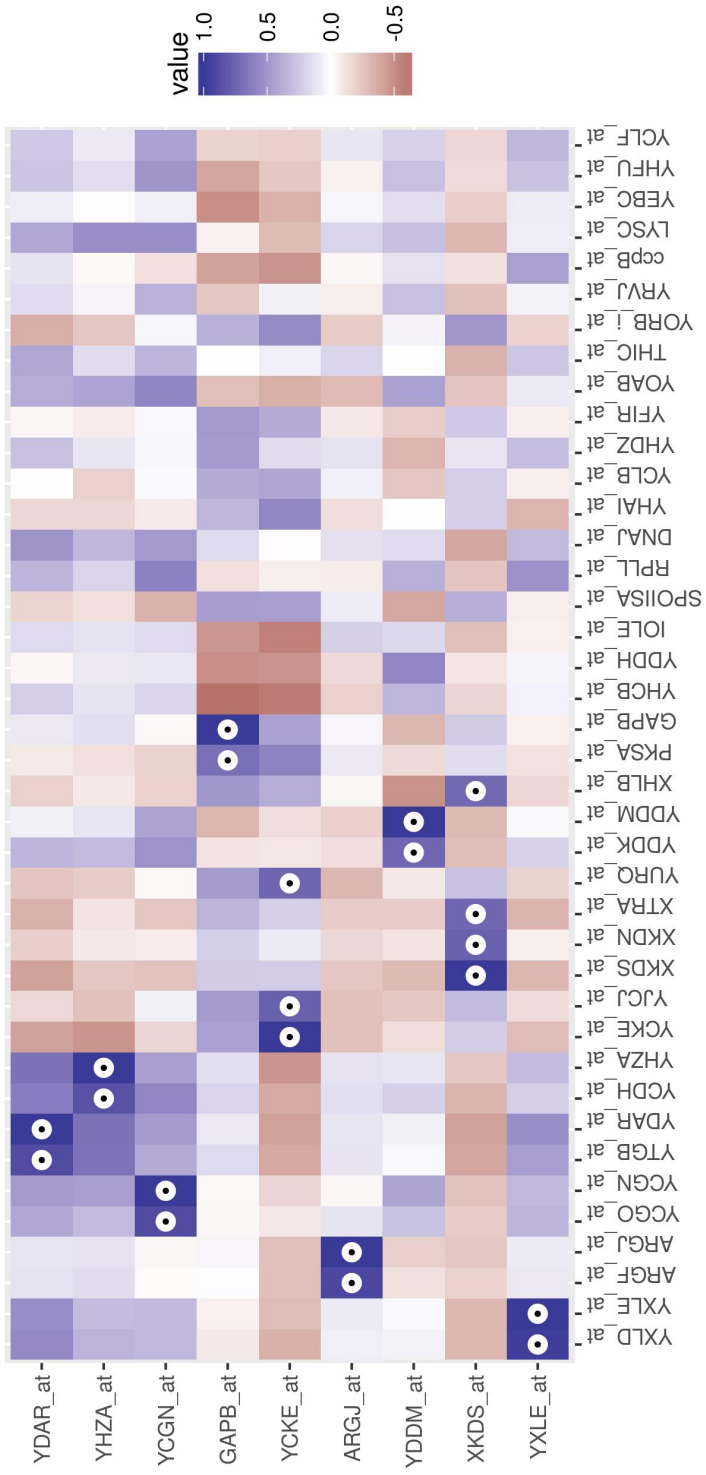

Figure 3.14: Correlation heat map for selected variables in riboflavin data set.

CHAPTER 4

LARON PACKAGE IN R

The `laron` package in R implements the LARON branching process described in Chapter 3, dropping potentially influential observations as it proceeds down the LARS path. This allows the user to explore predictor subspaces that might not be accessible when variable selection is performed with influential observations included. It also checks for variable collinearity that may be masked due to influential observations in minor eigenvector directions. It is assumed that data have not been normalized in advance, and therefore that the y-intercept must be estimated using $\bar{\mathbf{y}}$.

4.1 Main function

The main function performs the branching process and provides rankings for potential outliers and collinearity. The process may be implemented by calling the `laron` function as arguments. Given the simulated data described in Section 3.3.1 included in the package as `sim1` with predictor predictor matrix `x` and response vector `y`, the basic code operates as follows

```
> library(laron)
> data(sim1)
> x <- sim1[,-1]
> y <- sim1[,1]
> fit <- laron(x, y)
> fit
```

which results in the following output

```
Call:
laron.default(x = x, y = y)

 1 case nominated for investigation:
      Case                91
```

```
Avg Cook's Distance       367.357
   ----
 No evidence of collinarity.
```

The same output can be achieved by specifying a formula and data frame as arguments:

```
> fit <- laron(y ~ x, data = sim1)
```

By default, `laron` will fit a model with an intercept, assuming that the response variable has not been centered. To fit models without an intercept, either set the argument `intercept = FALSE` or, when specifying a formula, the code `y ~ x - 1` will remove the intercept. The resulting `fit` has associated `print`, `plot`, and `summary` functions for a natural user interaction with the results. The print function provides output like that shown above. Full lists of function arguments can be found in Appendix A.

The `summary` function gives further information concerning the outliers and any possible collinearity that may have been induced by the removal of any cases. There are three tables visible in the output.

The first gives further information on the influence value and selection ratio of individual cases. A subject's selection ratio indicates the proportion of branches in which the observation was included; A value close to 1 indicates that the removal of the point has a strong effect on the Lasso variable selection process. The influence measure may be reported as either a maximum or average over all branches and all steps. Influence measures have been normalized for comparison between models of different sizes.

The second table gives information on the condition number of the design matrix for the current model. Since we are iteratively removing cases during this

process, it is important to be aware of any induced or aggravated collinearity among predictors. Noting the branch and step number in which the condition numbers occur enable the user to identify cases potentially masking collinearity.

While condition numbers indicate potential numerical instability in the design matrix, VIF values indicate which variables are most likely to reduce collinearity by their removal. These values are given in the third table. Although the Lasso tends to arbitrarily select one of a correlated group of predictors and none of the others, a different active set may be selected by the same data with small changes. For more sensible comparison of variable selection results, it is advisable to keep only one predictor from a correlated group.

By default, nominations were performed using Chebychev probabilities on the average Cook's distance (for further details on the nomination procedure, see Section 4.2), and only subject 91 was selected. Although two other cases had fairly high selection ratios (0.688 and 0.562 respectively), they also had relatively small Cook's distances. There is no evidence of collinearity among these predictors, with a maximum VIF of only 1.4. Recall from Section 1.3 that condition numbers larger than 30 and VIFs larger than 5 are considered suspect.

```
> summary(fit)
Most Influential Cases:
   Subject Avg Cook's Distance Selection.Ratio
1:      91              367.357           0.938          *
2:      31                1.281           0.562
3:      71                1.072           0.688
---
Chebychev Prob:   0 *** 0.001 ** 0.010 * 0.050 .
   0.100   1

 Maximum Condition Numbers:
   Branch Step   Condition.Number
1:  9.000 12.000  2.005
2: 16.000 12.000  1.999
3: 12.000 12.000  1.995
---
Condition Numbers:   Inf *** 100 ** 50 * 30 . 15   0
```

```
 Most Collinear Variables:
   Variable Max.VIF
1:  x5      1.408
2:  x11     1.349
3:  x4      1.325
---
Variance Inflation Factors:   Inf ** 50 * 10 . 5   0
```

The results from the `laron` fit can also be plotted with `plot(fit)`. There are 9 possible diagnostic plots available in the `plot` function. By default, it will only show four of these plots: the residual sum of squares path, the residual correlation path, the maximum VIF, and the influence against the selection ratio. To see different plots, the `which.graphs` argument may be edited. For example, all nine plots were created using the following line of code:

```
> plot(fit, which.graphs = 1:9)
```

All plots for this data are shown in Figure 4.1.

4.1.1 Influence Measures and Cutoffs for Branching Process

The first step in the branching process by which laron explores the impact of observation removal occurs by nominating points with some influence measure greater than a given cutoff. The specification of the influence measure function and the cutoff value clearly have a significant impact on the behavior of the algorithm. In the `laron` function, these may be controlled via the `influenceMeasure`, `infcutoff`, and `infFUN` arguments. Note that the initial criteria for branch creation (described here) and nomination for investigation (in the Section 4.2) are distinct, as the secondary nomination is performed on influence measure summary statistics (an average or maximum) rather than in their raw form.

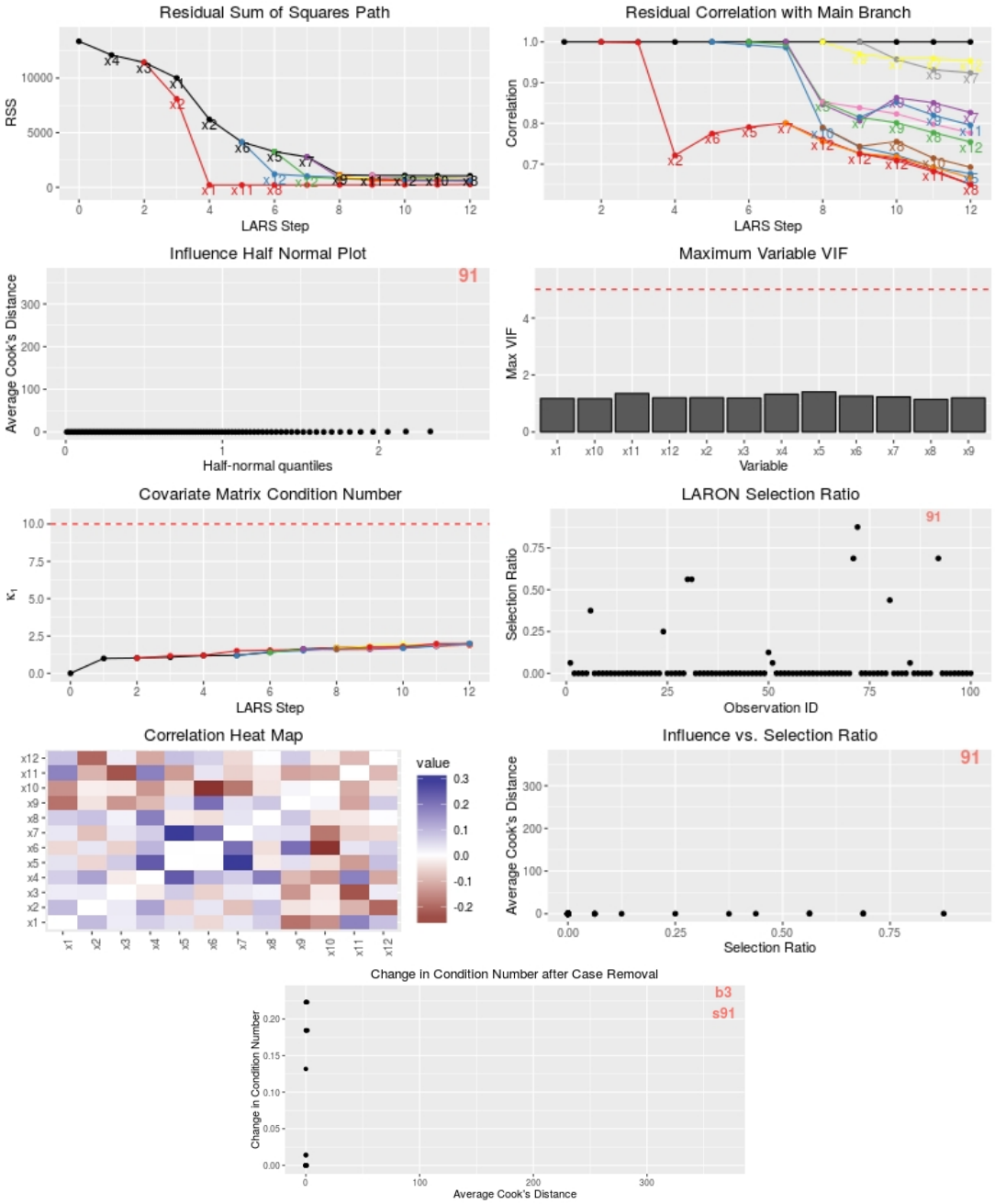

Figure 4.1: The `laron` package's diagnostic plots for the `sim1` data set. These plots are associated with `which.graph` numbers 1—9 when counting from left to right, then top to bottom.

There are three built-in influence measures available: Cook's distance, DFFITS, and DFBETAS, about which more information can be found in Section 1.2.2; these can be selected by setting `influenceMeasure = "cook"`, `"dffits"`, or `"dfbeta"`. Each built-in influence measure has a default cutoff point, set to $\frac{4}{n^*-p^*-1}$, $2\sqrt{\frac{p^*}{n^*}}$, and $\frac{2}{n^*}$ respectively, where the $*$ superscript denotes the effective sample size and active set size for the given branch and step. Cook's distance is set as the default.

Influence functions and cutoff values may also be entered manually. The `infFUN` argument may be passed a function taking as arguments the response vector `y` and predictor matrix `x`. As there are no default cutoff values for manual influence functions, it is necessary to set `infcutoff` as a function of `n` and `df` which represent the effective sample and active set sizes as in the previous paragraph. For example, if COVRATIO, an alternative influence measure described in [5], should be used, the following code will produce the desired result

```
> library(stats)
> fit2 <- laron(x, y, infFUN = function(y, x)
+     abs(covratio(lm(y ~ x)) - 1),
+     infcutoff = function(n, df) 3*df/n)
> fit2
Call:
laron.default(x = x, y = y, infFUN = function(y, x)
   abs(covratio(lm(y ~
    x)) - 1), infcutoff = function(n, df) 3 * df/n)

 1 case nominated for investigation:
    Case           91
Avg Influence     4.537
   ----
 No evidence of collinearity.
```

while the code below returns an error.

```
> fit2 <- laron(x, y, infFUN = function(y, x)
+     abs(covratio(lm(y ~ x)) - 1)
Error in laronMain(x, y, ...) :
  Must specify infcutoff when manually setting
     infFUN.
```

Note again that the cutoff identified here only pertains to observation nominations for the purpose of branch creation.

Many influence functions are not known to follow any probability distribution, however it is important to ensure that there are sufficient degrees of freedom available when creating a node for necessary calculations to be stable. For example, if the current branch with 25 subjects has already fit a model including 23 coefficients, it would be impossible to drop more than 2 cases to form a new branch. The default option controlling this is set to `min.df = 5`, derived from the connection between Cook's distance and the F distribution, which requires its second degree of freedom parameter be greater than or equal to 5 in order to maintain a finite variance.

4.1.2 Other Branching Options

While it is allowable for branches to break off from previous branches in this algorithm, it is wise to restrict the number of times this may occur. In sparse data particularly, where data tend to be decentralized, the algorithm may sequentially strip every observation it can. This option is controlled by the `max.level` option. "Level" denotes the number of breaks between the current branch and the "trunk" containing all observations (given level 0). Since selection rates are utilized in establishing outlier rankings, allowing the outlier selection to balloon can detract from the nominators effectiveness. This argument defaults to `max.level = max(3, ceiling(n/p))`.

Restrictions on the number of branches and the number of steps (arguments `max.steps` and `max.branch`, defaulting to `6*min(p,n-intercept)`

and `max(round(min(n,p)*(max.level))+1, 20)`, respectively) are extremely important to consider when working with very large data, since they will primarily control the amount of memory and time required to run the algorithm. The amount of information to be stored and used grows like the product `(max.steps + 1)·(max.branch)`. Particularly if you have extremely large data or limited access to RAM, you should ensure that the default values will not exceed your capacity. It is also not necessarily true that increasing the number or level of branches will yield better result; rather, if the values are increased too far it may produce spurious results. However, if they are set too low there may be a significant reduction in detection capacity. Given the general difficulty of dealing with high-dimensional data, it is not unlikely that relatively small changes the the branch options outlined here may yield different results.

4.1.3 Reading Branch Path Plots

The top row of plots in Figure 4.1 provides a visual glimpse into the branching process. These plots can be obtained by using the code `plot(fit, which.graphs = 1:2)`.

The first plot shows the LARS step plotted against the residual sum of squares (RSS) for the current model fit. The primary black line gives the LARS path for the full data, while the colored branches periodically breaking off from it (and sometimes from each other) indicate that a set of cases have been removed due to a sufficiently large influence measure and that their omission has changed the next variable to enter the model. Paths with significant reductions in RSS after branching off of the main data set indicate that among the omitted

points are likely outliers that should be reviewed by the analyst. This reduction need not take place immediately after breaking off. In the case of the simulated data `sim1`, the first red branch begins to deviate from the original at step 3 but does not reach its RSS minimum plateau until step 4, even though it correctly identifies the outlying observation (91). Such plateaus indicate that, after this point, the addition of new variables provides essentially no new information to the model. On the other hand, the full data model doesn't reach a minimum plateau until after four more variables have been added into the model. The reason in this data set is that case 91 is unusual in the 3 variables that enter the model in steps 5 through 7.

The second plot shows the deviation in the fitted values as observations are omitted. These deviations are accounted for by taking correlations of the predicted values from each branch with those of the original branch in the same step. In the case of the simulated data, it is clear that by step 4 the predictions between the original branch and the first red branch are quite different, even though both models have included the same four variables.

Both plots use the LARS step as the x-axis, indicating the number of variables in the model. Generally Lasso paths are indexed on a form of the penalty parameter such as the fractional form s for the LARS or the Lagrangian form λ for CD. Neither of these create easily interpretable paths when comparing models for different data. The Lagrangian values are not consistent across data sets; the fractional form, although it forms a more natural comparison point when examining models with different data, does not create an easily interpretable graph when plotted multiple data sets are plotted together; also, if the change in the full model is substantial due to the removal of a set of observations, the

fractional form may either exaggerate or mask the change in sparsity.

4.2 Outlier Nomination

It would not be advisable to recommend every observation nominated during the branching process, as this generally leads to unreasonably high false positive rates. Neither is it feasible to simply record all influence values and pick the largest, since influence measures tend to inflate as p increases. However, the influence values should provide important information for identifying extreme observations.

The second issue is addressed by "standardizing" the influence before storing it for later comparison. Storage occurs in two modes: running average and maximum influence values. Let D represent the vector of influence values, D^* the vector of values to be stored, and c be the cutoff value. At each step, the influence values are adjusted for storage such that

$$D^* = \frac{(D - c)^+}{3 \cdot \text{IQR}(D)}$$

where $(\cdot)^+ = \max(\cdot, 0)$. There are a number of ideas coalescing in this equation. Most influence measures are not associated with any probability distribution, so the cutoff value is generally the best available estimate for the expected change in D. Since it is also true that only those observations deemed sufficiently influential at a given point are of interest, it makes a reasonable pivot point. To justify the positive-part operator, it is important to note that an observation deemed particularly "uninfluential" at a given step in the process may simply be due to a key variable missing from the active set. Thus all observations not nominated are set to 0 for purposes of averaging rather than incorporating the negative

values. The IQR is used for scaling since there is no reason to believe that influence measures will behave symmetrically (rather, they are highly likely to be quite skewed), and Mirkin [51] recommends using 1.5·IQR(x) in place of $sd(x)$ for skewed data as a more stable measure of variation. Gelman [28] recommends dividing by $2 \cdot sd(x)$ for comparison purposes in regression situations, and so this is incorporated as well.

The main `laron` function automatically performs this outlier nomination process; however, this latter nomination may be changed without performing the time-consuming branch-building process again using the `outliers(fit,...)` function.

4.2.1 Nomination Criteria

Once all branches have been created, there is a collection of cases that have been removed to form new branches. This section addresses the options the user may specify to select the most interesting cases from among that group. All of the options described in this section may be used as arguments for the original `laron` call or the `outliers` function.

Instead of storing the influence measure of every observation at every step, `laron` only keeps track of the maximum and the average influence measures calculated during the process (after the standardization process described above). The `infType` argument can be used to specify which to use. The default is `infType = "average"` since the maximum is more likely to be unstable as p approaches n.

Other measures may also be used to nominate cases for investigation, controlled by the `nominator` argument. There are three general measures that could be passed into this argument: `"influence"` (the default),`"selection.ratio"` (or `"sr"`) which indicates the proportion of branches that omitted the observation, and `"ranker"` which is equal to the influence multiplied by the selection ratio plus one. The ranker appears to outperform the influence measure alone in high dimensional settings as a method for outlier nomination.

It is possible to control the cutoff point for nomination, which can be set manually or determined using one of three possible probability methods. The `cutoff` argument takes a vector of up to 4 cutoff values. These values are used to assign the relative significance of an observation. For example, we can assign specific cutoffs for the average Cook's distance for the LARON `fit` to the simulated data set `sim1` above.

```
> outliers(fit, cutoff = c(5, 10, 50, 100))
Most Influential Cases:
   Subject Avg Cook's Distance Selection.Ratio
1:      91             367.357           0.938          ***
2:      31               1.281           0.562
3:      71               1.072           0.688
---
Influence Measure: Inf *** 100 ** 50 * 10 . 5  0
```

Alternatively, we could determine that any observation with sufficiently large selection ratio (say, > 0.6) should be nominated for investigation. In this case, 4 observations are nominated instead of only 1.

```
> outliers(fit, nominator="sr", cutoff = c(0.6, 0.8,
    0.9))
Most Influential Cases:
   Subject  Avg Cook's Distance Selection.Ratio
1:      91             367.3574           0.938          **
2:      71               1.0716           0.688          .
3:      72               0.4527           0.875          *
4:      92               0.0967           0.688          .
---
```

```
Selection Ratio :   Inf ** 0.9 * 0.8 . 0.6   0
```

If only a single cutoff value is specified and the influence is used, relative significance is assigned according to whether only the maximum or both the average influence and the maximum exceed the cutoff.

```
> outliers.laron(fit, cutoff = 100, infType = "max")
Most Influential Cases:
   Subject Max Cook's Distance Selection.Ratio
1:   91    624.173              0.938          **
2:   47      4.670              0.000
3:   31      2.726              0.562
---
**  Average >  100
 *  Max >  100
```

An alternative is to determine cutoffs using probability inequalities. Most influence measures do not follow known probability distributions, so we rely instead on several theorems to make conservative nominations. Chebychev's inequality [40], the default option, provides a maximal tail probability for random variables with unknown distributions. Suppose random variable $X \sim \mathcal{F}$ where $\mathcal{F}$ is an unknown distribution with finite mean (μ) and variance (ς^2). Then Chebychev's inequality states

$$P\left(\frac{|X-\mu|}{\varsigma} \geq k\right) \leq \frac{1}{k^2}.$$

Another built-in option is to use Gaussian quantiles as cutoff points, though it is generally not recommended that this be used except with average influence measures which may reasonably be expected to behave Normally due to the wonders of the Central Limit Theorem.

It is important to note that all of these inequalities (as we have used them) rely on the assumption that all nominator values are drawn from the same probability distribution. Although this is likely not the case for the influence measures (which generally depend on the size of the design matrix to determine the

probability distribution) the standardization appears to make their distributions sufficiently similar as to allow these inequalities to work well in practice.

To select a probabilistic method in the `laron` or `outliers` function call, the argument `probType` can take values `"chebychev"` (the default) or `"normal"` for Gaussian quantile cutoffs. Partial matches are also acceptable. The probability values are by default set to `probs = c(0.001, 0.01, 0.05, 0.1)`, and can take only up to 4 values for significance levels.

4.2.2 Reading Outlier Plots

There are three outlier-specific graphs (visible in Figure 4.1) which can be obtained by `plot(laron.fit, which.graphs = c(3, 6, 8))` or `plot(outliers(laron.fit))`.

The first outlier plot gives the half-normal probability plot of the selected nominator value. Half-normal plots are commonly used for data that are known to be strictly positive, as here. Observations that are near the top right are considered suspect, particularly if they come after any large vertical gaps. Nominated points are labeled with their name or subject ID.

The second plot is a scatterplot of the selection ratio against the observation ID. In this plot, by default, observations are labeled if their selection ratio exceeds 0.9. Any point in the upper part of the graph may warrant further investigation.

The third plot examines the selection ratio and influence value simultaneously, and provides a visual representation of observations likely to be nomi-

nated using the "ranker" nominator. As in the first plot, points near the upper right corner ought to be investigated futher.

4.3 Collinearity

While the main loop is progressing, laron keeps track of two measures of collinearity: the condition number of the design matrix and VIFs for individual variables. These are recorded along the LARS path so that it is possible to identify potential causes of sudden spikes in collinearity. This may arise due to the entry of a collinear variable in the model or through the deletion of cases, as portrayed in Figure 3.1.

In data sets with no significant outliers and particularly in the presence of collinear predictors, laron tends to increase the number of cases that are removed in order to form a branch. When variables are correlated, the Lasso tends to select one of the group and ignore the others; changes in the data set may induce selection of a different variable from the group with relatively little impact on either the prediction or interpretability of the model.

To examine this phenomenon, let us utilize the diabetes data from the lars package

```
> fit3 <- laron(diabetes$x, log(diabetes$y))
> fit3
Call:
laron.default(x = diabetes$x, y = log(diabetes$y))

 9 cases nominated for investigation:
   Showing the first 5 cases.
       Case             93    388   290    59     381
Avg Cook's Distance  2.056  1.738  1.3  1.17  0.911
   ----
 Evidence of collinearity.
```

```
   Maximum condition number 23.71293
   Occurs in branch 4 step 10
   ----
 5 variables showed evidence of collinearity:
Variable    tc      ldl      hdl      ltg      tch
Max VIF   69.678   45.305   18.163   10.832   8.916
```

Here there are 9 cases nominated as outlying, and yet their average standardized Cook's distances are fairly small. By examining the `outliers` function, I can see that most of these points have a selection ratio of 0.75. Generally, I would not expect these observations to affect the set of predictors chosen by LARS. When variables `tc` and `tch` are removed (thus removing any indication of collinearity), the selection ratio of these observations drops to 0.5 or below. The second reason that so many observations are selected is due to the fact that the acceptable tail probability levels are set as high as 0.1 for outlier nomination. They are set this high intentionally as a conservative measure when the variance estimate is unnaturally inflated. Setting `probs = c(0.0001, 0.001, 0.01)` removes all but one case (93) from consideration.

Additional information on collinearity can be obtained through the `collinearity` function. This extracts information concerning high condition numbers (including the branch and step where it occurs) and VIFs. It also performs a check for groups of predictors with pairwise correlations above a certain threshold. This threshold `maxcor` is the only option available, and defaults to `0.8`. Using the `diabetes` data as an example again,

```
> collinearity(fit3, maxcor = 0.7)
 Largest Condition Numbers:
   Branch Step   Condition.Number
1:  4.000 10.000 23.713           .
2:  4.000  9.000 23.225           .
3:  2.000 10.000 22.465           .
4:  3.000 10.000 21.701           .
5:  1.000 12.000 21.682           .
6:  2.000  9.000 19.342           .
7:  1.000  9.000 18.826           .
```

```
8:  2.000  8.000 18.786           .
___
   Showing 8 of 12 nominated.
Condition Numbers:   Inf *** 100 ** 50 * 30 . 15   0

 Most Collinear Variables:
   Variable Max.VIF
1:   tc     69.678  **
2:  ldl     45.305  *
3:  hdl     18.163  *
4:  ltg     10.832  *
5:  tch      8.916  .
---
Variance Inflation Factors:   Inf ** 50 * 10 . 5   0

Correlated Groups:
Group 1:  tc  ldl
```

This example also provides an excellent reminder that collinearity and correlation are not synonymous in multiple regression. The `hdl`, `ltg`, and `tch` predictors exhibit high collinearity, though they are not correlated (with $r > 0.7$) with any other predictor.

4.3.1 Interpreting Collinear Graphs

There are four graphs that concern collinearity within the `laron` fit: a VIF bar chart, the condition number path, a correlation heat map, and a plot of the change in condition number against the influence of an omitted observation. When plotting a `laron` object, they are obtained with `which.graphs = c(4, 5, 7, 9)`. The first three may also be obtained by plotting the output from the `collinearity` function, with graph indices `1, 2`, and `3`, respectively. Examples of these plots for the `diabetes` data are shown in Figure 4.2.

The VIF bar chart shows the relative amount of collinearity exhibited by individual predictors. The condition number path allows for easy identification of the step in the LARS process where collinearity first becomes an issue for each

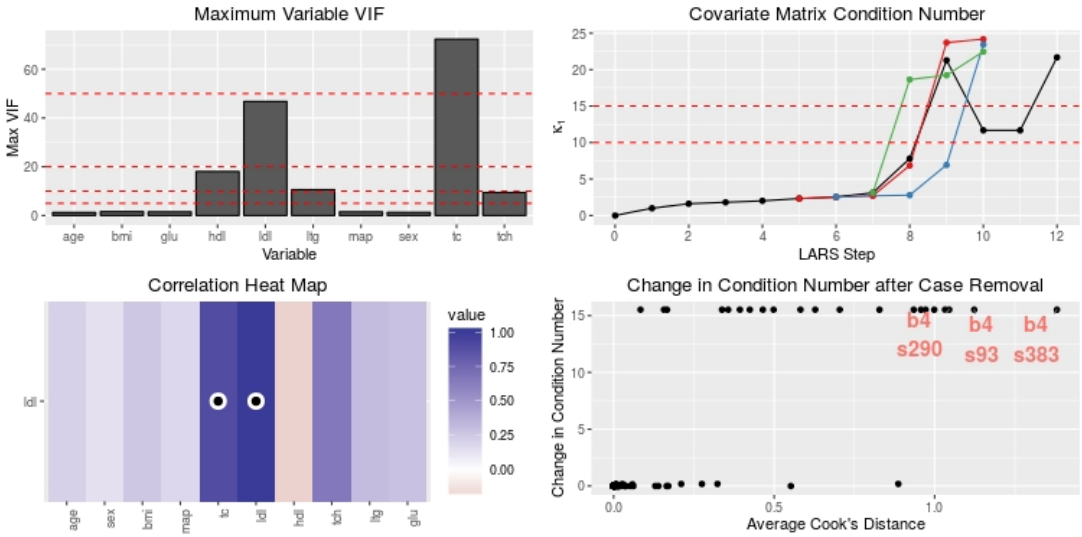

Figure 4.2: Collinearity plots for the `diabetes` data.

branch. Pairwise correlated variables are visible in the correlation heat map. If there are any correlated groups, the heatmap shows these across rows with dots indicating inclusion in the group. If there are no correlated groups, a standard correlation heat map is shown. The final plot indicates if there are any observations whose removal tends to uncover collinearity. Here, an observation's average influence is plotted against the absolute change in the condition number of the design matrix between the last step when it was included and the first step when the observation was removed. Cases with a large change (e.g. greater than 10) combined with a large influence, near the top right of the graph, may be suspicious. Points near the top right are labeled with their branch number and subject identification number for easy exploration.

In the case of the diabetes data shown in Figure 4.2, the VIF bar chart shows that predictors `tc` and `ldl` show the strongest evidence of collinearity. From the correlation heatmap, it is clear that the two variables are highly correlated with each other (r = 0.897). When `tc` is removed, only `tch` maintains its high VIF although the condition number stays below 15 for the entire path. In the

condition number path shown, all four branches eventually yield a high condition number, although only branch four has a steep spike immediately after it breaks off in step 7. This could indicate that the removal of these observations is the cause of the sudden spike; however, in this case it is unlikely, due to the low Cook's distances of the observations.

4.4 Branch Information

Branch information from the `laron` process can be obtained with the `branches` function. This function can be used to extract branch creation details, the list of observations removed for each branch, the order in which predictors enter the model for each branch, as well as RSS and residual correlation information. The basic function prints only the creation information and outlier selection for a subset of the first few branches.

```
> branches(fit)
Displaying 3 of 16 branches.

Branch Info:
           Parent Branch Split Step Level
Branch 1:               0          0     0
Branch 2:               1          3     1
Branch 3:               1          6     1

Outliers Selected by Branch:
Branch 1:
Branch 2:  91
Branch 3:  6  71 72 91 92
```

If we let `br <- branches(fit)`, the options for the associated print and summary functions can be used to obtain more information. For example

```
> summary(branches.laron(fit3))
Outliers Selected by Branch:
Branch 2:  30  57  59  78  93  124 157 170 260 290
   298 305 380 381 383 388 418
```

```
Branch 3:  30  57  59  78  93  103 124 157 212 260
   290 298 305 380 381 383 388 418
Branch 4:  30  57  59  78  93  103 111 124 157 170
   188 200 206 210 212 237 260 290 298 305 329 332
   354 364 380 381 383 388 418

Branch Variable Selection:
          Step1 Step2 Step3 Step4 Step5 Step6
Branch 1:  ltg   bmi   map   hdl   sex   tc
Branch 2:  ltg   bmi   map   hdl   sex   tc
Branch 3:  ltg   bmi   map   hdl   sex   tc
Branch 4:  ltg   bmi   map   hdl   sex   tc
          Step7 Step8 Step9 Step10 Step11 Step12
Branch 1:  glu   age   ldl   tch    -hdl    hdl
Branch 2:  ldl   glu   age   tch
Branch 3:  glu   ldl   age   tch
Branch 4:  ldl   glu   tch   age

Branch RSS Info:
          Min.RSS Min.RSS.Step Max.RSS.Drop
Branch 1:    71.0           12       -41.74
Branch 2:    55.7           10       -15.63
Branch 3:    54.5           10       -16.78
Branch 4:    49.0            9        -6.73
          RSS.Drop.Step
Branch 1:             2
Branch 2:             7
Branch 3:             8
Branch 4:             9
```

Further details on possible options may be found in Appendix A.0.9.

CHAPTER 5

CONCLUSION

5.1 Summary of findings

As with all other modeling schema, outliers can drastically affect the resulting model fit. In the case of the Lasso, outliers do not only impact prediction and coefficient estimation, but also the variable selection and sparsity of lasso estimation methods. Post-hoc analysis after fully fitting a model does not provide good detection capabilities, even in cases of relatively low dimensionality. Therefore it is clear that an alternative method should be utilized in order to determine which points are exerting unusual influence over the model building process.

The LARON algorithm has been shown to perform better than typical post-hoc analyses, although it also exhibits poor performance in the case of high leverage points in minor eigenvector directions. In situations of low dimensionality, outlier detection based solely on influence measures (either the maximum or the average) perform quite well. In medium and high dimensions, influence measures are inflated towards infinity uniformly for all observations and therefore do not perform well; however, scaling these values by the selection ratio (i.e. the proportion of times the observation's removal resulted in the selection of a different variable) outperforms the other methods tested.

This algorithm has been made available in an R package that performs outlier detection for linear models of any size and examines predictors for evidence of any collinearity, regardless of whether this collinearity has been masked by

the presence of unusual observations. This tool allows for more flexible and thorough exploration of the predictor space, as well as a visual glimpse into the effect of individual observations on the variable selection process.

In all cases, it is important to note that all diagnostic procedures should be approached with caution, as all are subject to issues of multiple comparisons and multiple tests. In the case of high dimensional data, we are subject to additional difficulties by limiting our examination to only a few of the many subspaces, selected through data-driven means. Any notable results derived from these diagnostics are meant to inform, not to exclude observations out of hand. There are certainly valid reasons for excluding variables or observations from model fits (such as input errors, unrecognized missing data codes, unusual and uncontrollable experimental environments, or high multicollinearity), however a positive diagnostic test is not sufficient to determine the appropriate action. If a point is found to be influential, why is it so? Where does it differ from other observations? Are there errors in the data that have not been addressed, or does it belong to a different subset of the population of interest? As with all data analysis, diagnostics much be approached with judgment; given the dirty world of data, there is no panacea for all situations, and the "correct" decision is often open to doubt.

5.2 Future work

There are two significant areas for improvement with the `laron` package: speed and memory usage. Even for reasonably small data, there can be a noticeable delay in the results; for large or high dimensional data sets, it may take several

hours to complete. Figure 5.1 indicates quite clearly that any addition to the dimensionality is computationally expensive. Although it appears from Figure 5.1 that the algorithm generally increases approximately linearly with respect to the number of variables, each additional variable costs approximately 12 seconds of computation time. Although the cost of an additional sample is less expensive with each additional observation adding only about half a second to the computation time, it appears that for $n > 5p$

The amount of information to be stored increases quite quickly, particularly as p grows. For example, in the `riboflavin` data set with a sample size of 71 and 4,088 predictors the resulting `laron` fit (with default settings) produced an output object occupying nearly 2 GB of space. This can be a very large concern for those with limited space in their working memory, and the necessity of reading and writing to virtual memory (located on the hard drive and significantly slower than working memory) can aggravate an already time-consuming process. Although this issue can be controlled to some extent by limiting the number of branches that may be formed, this also reduces the amount of available information to the analyst. For example, it is possible that an observation may not appear interesting until some obscure subspace of the predictor space is accessed, at which point the maximum number of branches may have already been created.

There are a number of ways that these issues might be addressed. Within the current algorithm, it may be possible to reduce the amount of information storage through the use of sparse matrix objects or more extensive use of integer indexing to reduce the amount of floating point storage. Decreasing the amount of memory required, and therefore the number of operations required, should

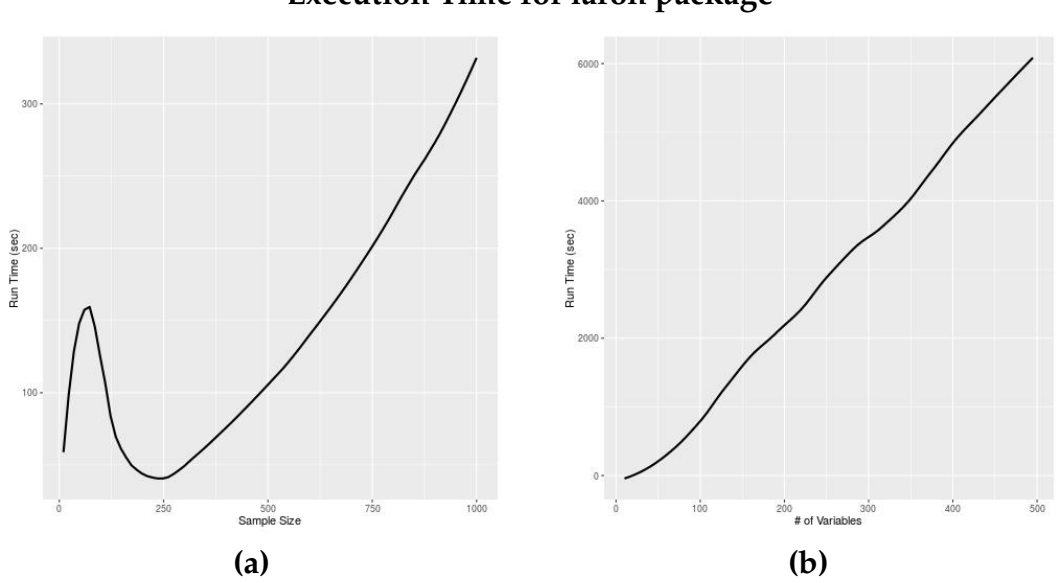

Figure 5.1: Execution time (in seconds) for laron package. The plot on the left varied the sample size with 50 variables; on the right, the sample size was fixed at 50 observations while the number of predictors varied. One influential outlier was generated in each case to ensure branches were formed.

also help to increase the computational speed. There may also be additional ways to vectorize within the program that have not yet been optimized.

Another possible method would be to iterate the CD algorithm on different subsets of data rather than the LARS algorithm, which is known to have improvements in speed. This would result in a loss of the branching tree process that provides a lot of visual information about how observations affect the data structure and the variable selection process, however it would also produce true full Lasso paths for each subset of observations.

The algorithm may also be updated to incorporate dummy variables for individual observations or sets of observations that are then eligible for selection within the process and follow the single path to see when and where these dummy variables are selected. There are several points of concern with this

strategy when performing a true Lasso fit. Since these dummy variables are in effect changing the value of the unpenalized (and unscaled) y-intercept, it is unreasonable to subject these coefficients to the same penalization as those of the variables of interest. Simply leaving all dummy variables unpenalized wastes precious degrees of freedom. In the `riboflavin` data, 49 cases (69%) were omitted in at least one branch. Implementing a secondary penalty parameter on these predictors appears to be the only option; since we have observed, however, that unusual observations can have a significant impact on the sparsity-inducing penalty parameter, it is quite likely that influential points will also alter the estimation procedure of this new penalty parameter.

Further research is also necessary in order to improve detection for high leverage points that are in low eigenvector directions. There may be interesting implications from the fact that the Lasso may be viewed as the limit of an iterated Ridge regression, described by Zou and Li [73]. Lichtenstein in her master's thesis [46] described how an observation's leverage in ridge regression can be expressed in terms of the singular value decomposition of the predictor matrix (as in standard regression) but with an added reliance on the Ridge penalty k. Assuming the SVD of $\mathbf{X} = UDV^T$, the Ridge leverage

$$h_i^{(Ridge)} = \sum_{j=1}^{p} \frac{d_j}{d_j + k} u_{ij}^2$$

is similar to the standard LS leverage $h_i = \sum_j u_{ij}^2$ but with a scaling factor dependent on the singular values of $\mathbf{X}$. This shows that, since k is fixed, observations with greater leverage distributed in the low-eigenvalue directions do not wield as much leverage in the overall ridge fit as those with high leverage in high-eigenvalue directions. It may be possible to combine the ideas of Lichtenstien, Fan, and Li to further explore the implications of leverage points in different

eigenvector directions on resulting Lasso fits, since it appears as though observations unusual in low eigenvalue directions are downweighted by Lasso in a similar manner.

APPENDIX A

COMPLETE OPTIONS FOR FUNCTIONS IN LARON PACKAGE

A.0.1 `laron` function

laron(x, y, influenceMeasure = c("cook","dffits","penalty.parameter","dfbeta"), infFUN, infcutoff, max.branch, min.df, max.level, intercept=TRUE, eps = .Machine$double.eps, max.steps, trace.step=FALSE, ...)

laron(y ~ x, data = *data.frame*, ...)

Returns an object of type `laron`

`x, y`

Predictor matrix and response vector

`influenceMeasure`

Defines which influence measure to useTakes one of the options `"cook","dffits","penalty.parameter",` or `"dfbeta"`.

`infFUN`

User-defined influence function. Must take arguments `y, x`. `infcutoff` must also be defined or will return error.

`infcutoff`

User-defined influence nomination cutoff point for branch nomination. Must take `n, p` as arguments.

`max.branch`

Integer determining the maximum number of branches that may be created by LARON.

min.df

Integer determining the minimum degrees of freedom that must exist for a new branch to be created. Default = 5.

max.level

Integer determining maximum level of branch splits.

max.steps

Integer determining the maximum number of LARS steps

intercept

Logical. If `TRUE` (the default) LARON fits a model with an intercept

eps Small number for catching small computational errors. Defaults to `.Machine$double.eps`

trace.step

Logical. If `TRUE`, LARON prints the LARS step. (Default: `FALSE`)

A.0.2 `outliers` function

outliers(laron.object, probs = c(0.001, 0.01, 0.05, 0.1), infType=c(”average”, ”maximum”), nominator=c(”influence”, ”selection.ratio”, ”sr”, ”ranker”), probType=c(”chebychev”, ”normal”), cutoff=NULL, ...)

Returns an object of type `outlier`.

probs

Numeric vector of length at most 4 containing tail probabilities for outlier nomination and relative significance levels.

infType

Which influence measure summary to use. One of ”average” or ”maximum”.

nominator

What type of measure to use to nominate subjects as outliers. One of influence”, ”selection.ratio”,”sr”, or ”ranker”.

probType

Type of probability for determining outlier nomination. One of ”chebychev” or ”normal”.

cutoff

Numeric vector of length at most 4 containing cutoff values for outlier nomination and relative significance levels. If specified, probabilities are not applied.

A.0.3 `collinearity` function

collinearity(laron.object, maxcor=0.8, ...)

Returns and object of type `multicol`.

maxcor

A value such that variables with pairwise correlations greater than maxcor are placed into groups.

A.0.4 `plot` function for `laron`, `outlier`, and `multicol` objects

plot(laron.object, which.graphs, max.label.step=20, max.plot.level=3, max.branch.plot=12, max.step.plot = 30, plot.points, caption, ask, branch.legend = FALSE, influenceType, textrepe, vlabs, slabs, maxp.plot = 15, label.pts, nlabel, group.heat=TRUE, ...)

plot(outlier.object, ...)

plot(multicol.object, ...)

`which.graphs`

Numeric vector containing elements from the set 1:9 (for a laron object). 1 = RSS; 2 = Correlation with original branch (same step); 3 = Influence half-normal plot (1 for outlier objects); 4 = VIF bar chart (1 for multicol objects); 5 = Condition Number path; 6 = Selection Ratio plot (2 for outlier objects); 7 = Pairwise correlation group heat map (2 for multicol object); 8 = Influence vs. Selection Ratio; 9 = Change in Condition Number

`max.label.step`

Integer determining the largest step to be labeled with variable label. Default = 20

`max.plot.level`

Integer specifying the largest branch level to be plotted. Default = 3

`max.branch.plot`

Integer specifying the most branches to plot on a graph. Default = 12

max.step.plot

Integer specifying the largest LARS step to plot. Default = 30

plot.points

Logical. Adds dots at each step in branching plots. Defaults to `TRUE` if `max.step.plot` $\leq$ `50`, else `FALSE`

caption

List of length 9 containing expressions of character strings for plot titles.

ask Logical. If `TRUE`, function asks for user to press <Enter> before progressing to the next plot. Defaults to `TRUE` if number of images per screen (as determinined by `par("mfrow")`) is less than the number to be displayed

branch.legend

Logical. Displays legend with branch colors if `TRUE`. Default = `FALSE`

influenceType

Influence measure to use for plotting. Defaults to value inherited from the `outlier` object. One of "average","maximum", "mean", "sr", "selection.ratio", "ranker")

textrepel

Logical. If `TRUE` (default if n >= 250 subject labels are repelled to ensure no overlap

vlabs

Variable labels. One of ("varnames" or "varids")

slabs

Subject labels. One of ("subnames", "subids")

maxp.plot

Maximum number of variables to show in a plot. Default = 15

label.pts

Numerical vector of subject indices to label on plot.

nlabel

Integer specifying the number of subjects to label in a plot.

group.heat

Logical. If `TRUE` (default) the correlation heat map will show groups of variables with high pairwise correlations.

A.0.5 `SRGrid` plotting function

SRGrid(x, ids, labelType = c(”id”, ”colname”), color =” darkblue”, label, title)

Prints a shaded grid of selection ratios for observations.

x May be an object of class `laron`, or a $b \times n$ matrix of values $\in [0, 1]$.

ids Vector of subject or column ids to include.

labelType

how categories in grid should be labeled. One of ”id”, ”dimname”, or a character vector of length n

color

Name of color for shading in graph.

title

Character string giving desired graph title (optional)

A.0.6 `summary` and `print` functions for `laron` objects

print(laron.object, digits=3, minshow=3, maxshow=5, concise = TRUE, ...)

summary.laron ¡- function(laron.object, ...)

`digits`

The number of digits to show in numerical summaries. Default = 3.

`minshow`

The minimum number of cases and predictors for which information should be printed. (Default = 3)

`maxshow`

The maximum number of cases and predictors for which information should be printed. (Default = 5 for print, 8 for summary).

`concise`

Logical; if TRUE (default), less information is shown.

`...` Additional arguments to the print and summary calls for outlier and multicol objects may be passed here.

A.0.7 `summary` and `print` functions for `outlier` objects

print(outlier.object, concise = FALSE, showall = ifelse(concise,FALSE,TRUE), digits=3, minshow=3, maxshow=ifelse(showall, x$np[1], ifelse(concise,5,8))

summary(outlier.object, showall = TRUE, ...)

concise

Logical. If FALSE (default) more information is shown.

showall

Logical. If TRUE (default) then information for all observations is shown.

digits

Integer specifying the number of digits to print for numerical summaries.

minshow

The minimum number of cases and predictors for which information should be printed. (Default = 3)

maxshow

The maximum number of cases and predictors for which information should be printed. (Default = 5 for print, 8 for summary).

A.0.8 `summary` and `print` functions for `multicol` objects

print(multicol.object, concise = FALSE, showall = FALSE, digits=3, minshow=3, maxshow = ifelse(showall, x$np[2], ifelse(concise,5,8)), showgps = ifelse(concise,FALSE,TRUE), vlabs = c(”varnames”, ”varids”),

summary.multicol ¡- function(x, showall = TRUE, ...)

concise

Logical. If FALSE (default) more information is shown.

showall

Logical. If TRUE (default) then information for all observations is shown.

digits

Integer specifying the number of digits to print for numerical summaries.

minshow

The minimum number of cases and predictors for which information should be printed. (Default = 3)

maxshow

The maximum number of cases and predictors for which information should be printed. (Default = 5 for print, 8 for summary).

showgps

Logical. If TRUE, highly pairwise correlated group lists are printed.

vlabs

Variable labels. One of (”varnames” or ”varids”)

A.0.9 `summary` and `print` functions for `branch` objects

br <- branches(*laron.object*)

print(br, max.level=3, max.branch = ifelse(concise, 5, 12), max.step = ifelse(concise, 6, 20), include = c(”first”,”hasNominee”,”all”), which.branches, any.obs, all.obs, slabs = c(”subnames”, ”subids”), vlabs = c(”varnames”, ”varids”, ”none”), concise = TRUE, infoType = c(”overview”,”outliers”,”actions”, ”rss”, ”correlation”, ”all”), digits=3, ...)

summary(br, ...)

max.level

The largest level of a branch to print (default = 3). May be overruled by

include

max.branch

The maximum number of branches for which to print information. Overruled when include = "all".

all.obs,

any.obs

Takes avector of subject indices as an argument. Prints information of branches including all or any of the observations listed, respectively.

slabs

Takes value "subnames" (default) or "subids" to print subject information using their names (row names for the x matrix) or row ID numbers.

vlabs

Takes values ”varnames” (default) or "varids" to print variable information using variable names (column names of x matrix) or

APPENDIX B

ADDITIONAL INFORMATION ON SIMULATION RESULTS AND THE RIBOFLAVIN DATA

The set of tables (B.1 thru B.6) are provide further information on false alarm and detection rates as different aspects of the outlier behavior change. All data were generated according to the simulation description in Section 3.2.1, with the changing elements listed in the table. These tables provide insight that is unavailable from the ROC curves concerning the appropriateness of the cutoff levels utilized in the algorithm.

Based on these tables, it is clear that the FAR tends to decrease as the residual value goes up. The high rate of false alarms with relatively little influence is due to the fact that cutoffs are conservatively set to be based on the variance of the observed influence values resulting from the LARON run. FARs are also tend toward toward the middle as the contamination level increases: it selects fewer false alarms when the residual is high, but more when the residual is low compared to data sets with fewer outliers. Regardless of the situation, the detection rate is extremely good for residual points.

These cutoffs provide good detection capabilities and vastly smaller FARs for high leverage points as well, however it does not perform as well at detecting all outliers when more than one outlier has been introduced in to the data.

Additional information on the riboflavin data set can be found in Tables B.7 and B.8.

Table B.1: High Residual Outliers: False Alarms

$\lvert\mathcal{O}\rvert$	δ_y	FA	sd	# FAs	sd	$\frac{\text{\# FAs}}{\text{\# Nominated}}$	sd
1	4.30	0.93	0.26	2.63	1.31	0.66	0.23
1	7.00	0.51	0.50	1.73	0.96	0.30	0.31
1	39.62	0.01	0.10	1.00		0.01	0.05
2	4.30	0.82	0.39	2.48	1.34	0.52	0.30
2	7.00	0.43	0.50	1.42	0.59	0.19	0.24
2	39.62	0.07	0.26	1.00	0.00	0.03	0.09
5	4.30	0.82	0.39	1.98	0.96	0.39	0.24
5	7.00	0.53	0.50	1.38	0.60	0.16	0.16
5	39.62	0.13	0.34	1.15	0.38	0.04	0.10

Table B.2: High Residual Outliers: Detection Rates

$\lvert\mathcal{O}\rvert$	δ_y	DR	sd	DR_{all}	sd	# $\mathcal{O}$ Det	sd	$\frac{\text{\# Detected}}{\text{\# Outliers}}$	sd
1	4.30	0.91	0.29	0.91	0.29	1.00	0.00	0.91	0.29
1	7.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1	39.62	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
2	4.30	0.92	0.27	0.42	0.50	1.46	0.50	0.67	0.31
2	7.00	1.00	0.00	0.67	0.47	1.67	0.47	0.83	0.24
2	39.62	1.00	0.00	0.93	0.26	1.93	0.26	0.96	0.13
5	4.30	0.97	0.17	0.01	0.10	2.46	0.90	0.48	0.20
5	7.00	1.00	0.00	0.13	0.34	3.35	0.97	0.67	0.19
5	39.62	1.00	0.00	0.09	0.29	2.86	1.21	0.57	0.24

Table B.3: High Residual Outliers: Outlier Info

$\lvert\mathcal{O}\rvert$	δ_y	Level	sd	Step	sd	$\text{Rank}_{\mathcal{O}}$	sd	$D_{\mathcal{O}}$	sd	$\text{SR}_{\mathcal{O}}$	sd	$D_{\mathcal{O}^C}$	sd	$\text{SR}_{\mathcal{O}^C}$	sd
1	4.30	1.79	1.79	0.22	0.11	1.03	0.36	5.75	1.85	0.168	0.075	0.028	0.003	0.008	0.0019
1	7.00	1.04	0.20	0.48	0.26	0.99	0.22	5.68	1.53	0.233	0.094	0.027	0.003	0.007	0.0016
1	39.62	1.00	0.00	1.43	0.67	1.00	0.00	3.93	1.78	0.497	0.171	0.027	0.004	0.005	0.0017
2	4.30	9.10	13.38	0.18	0.12	0.98	0.48	5.26	2.12	0.152	0.082	0.028	0.003	0.007	0.0017
2	7.00	2.68	1.74	0.33	0.20	1.05	0.30	5.56	1.46	0.216	0.090	0.027	0.003	0.006	0.0017
2	39.62	2.01	0.10	0.65	0.30	1.00	0.00	3.33	1.90	0.292	0.092	0.026	0.004	0.004	0.0019
5	4.30	11.20	4.31	0.13	0.06	1.09	0.39	5.60	1.39	0.118	0.045	0.028	0.004	0.004	0.0016
5	7.00	6.99	1.82	0.19	0.08	1.03	0.23	5.48	1.18	0.135	0.032	0.028	0.003	0.003	0.0015
5	39.62	5.85	1.32	0.32	0.30	1.02	0.15	2.82	1.74	0.142	0.026	0.026	0.005	0.003	0.0014

Table B.4: Leverage Outliers: False Alarms

$\lvert\mathcal{O}\rvert$	j	δ_x	FA	sd	# FAs	sd	$\frac{\text{\# FAs}}{\text{\# Nominated}}$	sd
1	major	0.632	0.05	0.22	3.20	2.05	0.03	0.16
1	major	1.190	0.02	0.14	3.00	1.41	0.01	0.10
1	major	2.956	0.00	0.00			0.00	0.00
1	minor	0.632	0.14	0.35	2.50	1.74	0.10	0.27
1	minor	1.190	0.06	0.24	2.67	1.03	0.04	0.17
1	minor	2.956	0.00	0.00			0.00	0.00
2	major	0.632	0.00	0.00			0.00	0.00
2	major	1.190	0.00	0.00			0.00	0.00
2	major	2.956	0.00	0.00			0.00	0.00
2	minor	0.632	0.03	0.17	2.00	1.73	0.02	0.10
2	minor	1.190	0.01	0.10	2.00		0.01	0.07
2	minor	2.956	0.00	0.00			0.00	0.00
5	major	0.632	0.00	0.00			0.00	0.00
5	major	1.190	0.00	0.00			0.00	0.00
5	major	2.956	0.00	0.00			0.00	0.00
5	minor	0.632	0.00	0.00			0.00	0.00
5	minor	1.190	0.00	0.00			0.00	0.00
5	minor	2.956	0.00	0.00			0.00	0.00

Table B.5: Leverage Outliers: Detection Rates

$\lvert\mathcal{O}\rvert$	j	δ_x	DR	sd	DR_{all}	sd	# $\mathcal{O}$ Det	sd	$\frac{\text{\# Detected}}{\text{\# Outliers}}$	sd
1	major	0.632	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1	major	1.190	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1	major	2.956	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1	minor	0.632	0.96	0.20	0.96	0.20	1.00	0.00	0.96	0.20
1	minor	1.190	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
1	minor	2.956	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
2	major	0.632	1.00	0.00	0.41	0.49	1.41	0.49	0.70	0.25
2	major	1.190	1.00	0.00	0.34	0.48	1.34	0.48	0.67	0.24
2	major	2.956	1.00	0.00	0.47	0.50	1.47	0.50	0.73	0.25
2	minor	0.632	1.00	0.00	0.27	0.45	1.27	0.45	0.64	0.22
2	minor	1.190	1.00	0.00	0.30	0.46	1.30	0.46	0.65	0.23
2	minor	2.956	1.00	0.00	0.46	0.50	1.46	0.50	0.73	0.25
5	major	0.632	1.00	0.00	0.00	0.00	1.82	0.76	0.36	0.15
5	major	1.190	1.00	0.00	0.00	0.00	1.89	0.79	0.38	0.16
5	major	2.956	1.00	0.00	0.00	0.00	1.62	0.68	0.32	0.14
5	minor	0.632	1.00	0.00	0.00	0.00	1.94	0.84	0.39	0.17
5	minor	1.190	1.00	0.00	0.00	0.00	1.90	0.72	0.38	0.14
5	minor	2.956	1.00	0.00	0.00	0.00	1.65	0.66	0.33	0.13

Table B.6: Leverage Outliers: Outlier Info

$\lvert\mathcal{O}\rvert$	j	δ_x	Level	sd	Step	sd	$\text{Rank}_{\mathcal{O}}$	sd	$D_{\mathcal{O}}$	sd	$\text{SR}_{\mathcal{O}}$	sd	$D_{\mathcal{O}^C}$	sd	$\text{SR}_{\mathcal{O}^C}$	sd
1	major	0.632	1.03	0.17	3.67	1.61	0.97	0.22	4.57	1.66	0.36	0.220	0.028	0.000	0.006	0.002
1	major	1.190	1.00	0.00	13.72	5.76	1.00	0.20	3.91	1.81	0.22	0.118	0.029	0.004	0.007	0.001
1	major	2.956	1.00	0.00	85.86	26.50	1.00	0.00	2.46	1.61	0.17	0.048	0.033	0.004	0.008	0.000
1	minor	0.632	1.78	3.97	2.37	1.44	1.00	0.57	5.39	2.18	0.30	0.233	0.027	0.003	0.006	0.002
1	minor	1.190	1.04	0.28	8.84	4.99	0.89	0.42	4.35	2.57	0.18	0.122	0.028	0.003	0.006	0.002
1	minor	2.956	1.00	0.00	56.00	32.60	0.94	0.28	3.78	2.46	0.17	0.088	0.029	0.004	0.007	0.002
2	major	0.632	3.06	6.06	3.57	3.42	1.03	0.27	4.90	1.71	0.23	0.127	0.027	0.003	0.005	0.001
2	major	1.190	2.49	4.31	11.12	10.62	1.03	0.24	3.89	1.86	0.17	0.064	0.029	0.004	0.006	0.001
2	major	2.956	2.00	0.00	63.32	53.98	1.02	0.14	2.53	1.49	0.17	0.042	0.032	0.004	0.006	0.000
2	minor	0.632	6.86	15.06	1.11	1.11	1.03	0.45	5.14	1.86	0.25	0.158	0.026	0.003	0.004	0.002
2	minor	1.190	2.25	1.51	4.22	4.06	1.04	0.23	4.85	1.57	0.22	0.081	0.028	0.003	0.005	0.001
2	minor	2.956	2.00	0.00	26.11	21.98	1.06	0.29	3.04	1.60	0.15	0.042	0.032	0.004	0.007	0.001
5	major	0.632	11.36	17.11	1.62	2.56	1.26	0.64	4.76	2.25	0.13	0.074	0.026	0.004	0.003	0.001
5	major	1.190	6.57	9.07	4.86	7.70	1.25	0.56	3.52	2.05	0.10	0.040	0.030	0.004	0.004	0.001
5	major	2.956	5.64	1.93	25.16	50.77	1.34	0.54	3.16	1.55	0.10	0.032	0.038	0.006	0.005	0.000
5	minor	0.632	22.62	27.70	0.65	0.92	1.06	0.66	4.69	2.66	0.14	0.099	0.023	0.004	0.002	0.001
5	minor	1.190	13.53	21.35	2.10	2.97	1.20	0.59	4.40	2.34	0.12	0.063	0.027	0.004	0.004	0.001
5	minor	2.956	6.52	5.64	9.50	16.07	1.27	0.53	3.45	1.83	0.11	0.041	0.031	0.004	0.004	0.001

Table B.7: Full table of coefficient values for riboflavin model fits.

Lasso 1SE		TREX		EB		−{29, 31, 51}		−{29, 31, 51, 59}	
(Intercept)	1.155	Intercept	-7.159	(Intercept)	-7.600	(Intercept)	-5.202	(Intercept)	-8.106
YOAB_at	-0.697	YXLD_at	-0.219	YOAB_at	-1.130	SPOIISA_at	0.322	SPOIISA_at	0.298
YEBC_at	-0.428	YOAB_at	-0.168	YXLE_at	-0.920	YEBC_at	-0.289	XKDS_at	0.185
LYSC_at	-0.285	ARGF_at	-0.112	ARGF_at	-0.860	YXLE_at	-0.203	YCKE_at	0.178
YXLD_at	-0.212	YEBC_at	-0.088	XHLB_at	0.690	YDDM_at	-0.166	YXLE_at	-0.133
YBFI_at	0.178	YCKE_at	0.069	XKDN_at	-0.510	THIC_at	-0.164	YDAR_at	-0.124
XHLB_at	0.175	YCGO_at	-0.065	YHDZ_at	0.460	YTGB_at	-0.125	YHCB_at	-0.095
YDDK_at	-0.158	YEZB_at	0.049	XKDS_at	0.430	YJCJ_at	0.125	YCGN_at	-0.085
YCLB_at	0.157	YFHE_r_at	0.041			YCKE_at	0.105	YOAB_at	-0.084
DNAJ_at	-0.148	YHZA_at	-0.030			XTRA_at	0.101	YTGB_at	-0.046
ARGF_at	-0.145	LYSC_at	-0.024			YFHE_r_at	0.098	XTRA_at	0.026
YURQ_at	0.120	YDDK_at	-0.024			YCGN_at	-0.098	YJCJ_at	0.022
YFHE_r_at	0.104	RPLL_at	-0.022			YPGA_at	-0.091	YXLJ_at	-0.014
SPOVAA_at	0.099	YXLE_at	-0.019			XKDS_at	0.088	YORB_i_at	0.011
YEZB_at	0.079	YCDH_at	-0.015			YORB_i_at	0.079	YDBM_at	-0.006
YXLE_at	-0.070	YYDA_at	-0.015			IOLE_at	-0.072	XHLA_at	0.005
YCKE_at	0.068	YBFI_at	0.007			YHAI_at	0.058	YXLD_at	-0.005
PKSA_at	0.065	YHDS_r_at	0.006			CTAA_at	0.052	YHFU_at	-0.002
YYDA_at	-0.065	PKSA_at	0.003			YOAB_at	-0.039	XHLB_at	0.000
YHDS_r_at	0.056	SPOVAA_at	0.003			ccpB_at	-0.026		
XTRA_at	0.045	YDDH_at	-0.001			ARGF_at	-0.024		
YQJU_at	0.037					YHCB_at	-0.024		
YCLF_at	-0.034					YCLB_at	0.021		
YFIR_at	0.034					YKBA_at	0.020		
YRVJ_at	-0.029					YCGO_at	-0.013		
XKDS_at	0.028					YVFK_at	0.011		
YKBA_at	0.028					YXLD_at	-0.011		
SPOIISA_at	0.023					ARGJ_at	-0.000		
YCGO_at	-0.015								
GAPB_at	0.009								
YDDH_at	-0.006								
YCDH_at	-0.000								

Table B.8: List of case names for riboflavin data.

Case ID	Case Name
1	b_Fbat107PT24.CEL
2	b_Fbat107PT30.CEL
3	b_Fbat107PT48.CEL
4	b_Fbat107PT52.CEL
5	knh_102_Fbat289PT24.CEL
6	knh_103_Fbat289PT48.CEL
7	knh_490_BS0001_Fbat1070_PT24_1.Rep.CEL
8	knh_491_BS0001_Fbat1070_PT30_1.Rep.CEL
9	knh_492_BS0001_Fbat1070_PT48_1.Rep.CEL
10	knh_494_BS0001_Fbat1077_PT24_2.Rep.CEL
11	knh_495_BS0001_Fbat1077_PT30_2.Rep.CEL
12	knh_496_BS0001_Fbat1077_PT48_2.Rep.CEL
13	knh_498_BS0001_Fbat1091_PT24_3.Rep.CEL
14	knh_499_BS0001_Fbat1091_PT30_3.Rep.CEL
15	knh_500_BS0001_Fbat1091_PT48_3.Rep.CEL
16	knh_502_BS3416_E_Fbat1071_PT24_1.Rep.CEL
17	knh_503_BS3416_E_Fbat1071_PT30_1.Rep.CEL
18	knh_504_BS3416_E_Fbat1071_PT48_1.Rep.CEL
19	knh_506_BS3416_E_Fbat1078_PT24_2.Rep.CEL
20	knh_507_BS3416_E_Fbat1078_PT30_2.Rep.CEL
21	knh_508_BS3416_E_Fbat1078_PT48_2.Rep.CEL
22	knh_510_BS3416_E_Fbat1092_PT24_3.Rep.CEL

Continued on next page

Table B.8 – *Continued from previous page*

Case ID	Case Name
23	knh_512_BS3416_E_Fbat1092_PT48_3.Rep.CEL
24	knh_532_BS3416_E_Fbat1092_PT30_3.Rep_wh.CEL
25	knh_655_BS5009_Fbat1374_PT24_1.Repl._No.2_repeated RNA.CEL
26	knh_657_BS5009_Fbat1379_PT24_2.Repl._No.6_repeated RNA.CEL
27	knh_659_BS5009_Fbat1383_PT24_3.Repl._No.10_repeated RNA.CEL
28	knh_660_BS5009_Fbat1374_PT30_1.Repl._No.3_rep_cDNA_new Yeast.CEL
29	knh_661_BS5009_Fbat1374_PT48_1.Repl._No.4_rep_cDNA_new Yeast.CEL
30	knh_662_BS5009_Fbat1379_PT30_2.Repl._No.7_rep_cDNA_new Yeast.CEL
31	knh_663_BS5009_Fbat1379_PT48_2.Repl._No.8_rep_cDNA_new Yeast.CEL
32	knh_664_BS5009_Fbat1383_PT30_3.Repl._No.11_rep_cDNA_new Yeas.CEL
33	knh_665_BS5009_Fbat1383_PT48_3.Repl._No.12_rep_cDNA_new Yeas.CEL
34	knh_679_BS5254_Fbat1442_PT24_1.Repl._No.38.CEL
35	knh_680_BS5254_Fbat1442_PT30_1.Repl._No.39.CEL
36	knh_681_BS5254_Fbat1442_PT48_1.Repl._No.40.CEL
37	knh_683_BS5254_Fbat1443_PT24_2.Repl._No.42.CEL
38	knh_684_BS5254_Fbat1443_PT30_2.Repl._No.43.CEL
39	knh_685_BS5254_Fbat1443_PT48_2.Repl._No.44.CEL
40	knh_687_BS5254_Fbat1444_PT24_3.Repl._No.46.CEL
41	knh_688_BS5254_Fbat1444_PT30_3.Repl._No.47.CEL
42	knh_689_BS5254_Fbat1444_PT48_3.Repl._No.48.CEL
43	knhb_039_Fbat152PT24.CEL
44	knhb_040_Fbat152PT48.CEL
45	knhb_041_Fbat152PT52.CEL

Continued on next page

Table B.8 – *Continued from previous page*

Case ID	Case Name
46	knhb_046_Fbat202PT24.CEL
47	knhb_047_Fbat202PT48.CEL
48	knhb_049_Fbat203PT24.CEL
49	knhb_050_Fbat203PT48.CEL
50	knhb_090_Fbat284PT24.CEL
51	knhb_091_Fbat284PT48.CEL
52	knhb_098_Fbat287PT24.CEL
53	knhb_099_Fbat287PT48.CEL
54	knhb_169_Fbat395PT24.CEL
55	knhb_170_Fbat395PT48.CEL
56	knhb_174_Fbat396PT24.CEL
57	knhb_175_Fbat396PT48.CEL
58	knhb_180_Fbat397PT24.CEL
59	knhb_181_Fbat397PT48.CEL
60	knhb_186_Fbat398PT24.CEL
61	knhb_187_Fbat398PT48.CEL
62	knhb_237_Fbat394PT24.CEL
63	knhb_238_Fbat394PT48.CEL
64	knhb_247_Fbat525PT24.CEL
65	knhb_248_Fbat525PT48.CEL
66	knhb_251_Fbat526PT24.CEL
67	knhb_252_Fbat526PT48.CEL
68	knhb_255_Fbat527PT24.CEL

Continued on next page

Table B.8 – *Continued from previous page*

Case ID	Case Name
69	knhb_256_Fbat527PT48.CEL
70	knhb_259_Fbat528PT24.CEL
71	knhb_260_Fbat528PT48.CEL

REFERENCES

[1] Charu Aggarwal and S Yu. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB JournalThe International Journal on Very Large Data Bases*, 14(2):211–221, 2005.

[2] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–27. Springer, 2002.

[3] F. Bach, R. Jennaton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2011.

[4] Haim Y Bar, James G Booth, and Martin T Wells. A scalable empirical bayes approach to variable selection. *arXiv preprint arXiv:1510.03781*, 2015.

[5] David A Belsley, Edwin Kuh, et al. *Regression diagnostics*. Wiley, NY, 1980.

[6] SM Bendre, V Barnett, and T Lewis. *Outliers in Statistical Data*. Wiley, NY, 1994.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[8] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC Press, 1984.

[9] Peter Bühlmann, Markus Kalisch, and Lukas Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Applications*, 2014.

[10] Florentina Bunea, Alexandre Tsybakov, Marten Wegkamp, et al. Sparsity oracle inequalities for the lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

[11] Beth L. Chance. *Behavior characterization and estimation for general hierarchical multivariate linear regression models*. PhD thesis, Cornell University, Ithaca, NY, 1994.

[12] Samprit Chatterjee and Ali S Hadi. *Sensitivity analysis in linear regression*, volume 327. John Wiley & Sons, 2009.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.

[14] Stuart Coles, Joanna Bawa, Lesley Trenner, and Pat Dorazio. *An introduction to statistical modeling of extreme values*, volume 208. Springer, 2001.

[15] Gabriela Czanner, Sridevi V Sarma, Uri T Eden, and Emery N Brown. A signal-to-noise ratio estimator for generalized linear model systems. In *Proceedings of the World Congress on Engineering*, volume 2, 2008.

[16] Arnak S Dalalyan, Mohamed Hebiri, and Johannes Lederer. On the prediction performance of the lasso. *arXiv preprint arXiv:1402.1700*, 2014.

[17] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[18] Norman R Draper and R Craig Van Nostrand. Ridge regression and jamesstein estimation: review and comments. *Technometrics*, 21(4):451–466, 1979.

[19] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[20] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[21] Jianqing Fan, Yingying Fan, and Jinchi Lv. High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, 147(1):186–197, 2008.

[22] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[23] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

[24] Julian J Faraway. *Linear Models with R*. CRC Press, 2014.

[25] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[26] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[27] Pierre Garrigues and Laurent El Ghaoui. An homotopy algorithm for the lasso with online observations. In *NIPS*, pages 489–496, 2008.

[28] Andrew Gelman. Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine*, 27(15):2865–2873, 2008.

[29] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[30] Ali S Hadi and Jeffrey S Simonoff. Procedures for the identification of multiple outliers in linear models. *Journal of the American Statistical Association*, 88(424):1264–1272, 1993.

[31] Chris Hans. Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 106(496):1383–1393, 2011.

[32] Trevor J.. Hastie, Robert John Tibshirani, and Jerome H Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2011.

[33] Mohamed Hebiri and Johannes Lederer. How correlations influence lasso prediction. *Information Theory, IEEE Transactions on*, 59(3):1846–1854, 2013.

[34] Nicholas J Higham. *Accuracy and Stability of Numerical Algorithms*. Siam, 2002.

[35] Victoria J Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[36] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[37] Peter J Huber. *Robust statistics. 1981*. Wiley, New York.

[38] David R Hunter and Runze Li. Variable selection using mm algorithms. *The Annals of Statistics*, 33(4):1617, 2005.

[39] Eamonn Keogh and Abdullah Mueen. Curse of dimensionality. In *Encyclopedia of Machine Learning*, pages 257–258. Springer, 2011.

[40] G Jay Kerns. *Introduction to Probability and Statistics Using R*. Lulu.com, 2010.

[41] Seung-Jean Kim, Kwangmoo Koh, Michael Lustig, Stephen Boyd, and Dimitry Gorinevsky. An interior-point method for large-scale-regularized least squares. *IEEE journal of selected topics in signal processing*, 1(4):606–617, 2007.

[42] Minjung Kyung, Jeff Gill, Malay Ghosh, George Casella, et al. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.

[43] Johannes Lederer and Christian Müller. Don't fall for tuning parameters: tuning-free variable selection in high dimensions with the trex. *arXiv preprint arXiv:1404.0541*, 2014.

[44] Friedrich Leisch. Creating r packages: A tutorial. 2008.

[45] Qing Li, Nan Lin, et al. The bayesian elastic net. *Bayesian Analysis*, 5(1):151–170, 2010.

[46] Carolyn Helen Lichtenstein. Ridge regression and its effect on high leverage points in the data. Master's thesis, Cornell University, Ithaca, NY, 1981.

[47] Dmitry M Malioutov, Müjdat Cetin, and Alan S Willsky. Homotopy continuation for sparse signal representation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–733. IEEE, 2005.

[48] Donald W Marquaridt. Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics*, 12(3):591–612, 1970.

[49] Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007.

[50] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

[51] Boris Mirkin. *Clustering: a data recovery approach*. CRC Press, 2012.

[52] Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

[53] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326. IEEE, 2003.

[54] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[55] Nicholas G Polson and James G Scott. Local shrinkage rules, lévy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):287–311, 2012.

[56] Nicholas G Polson and James G Scott. Mixtures, envelopes, and hierarchical duality. *arXiv preprint arXiv:1406.0177*, 2014.

[57] Jean-Louis Rigal and Jean Gaches. On the compatibility of a given solution with the data of a linear system. *Journal of the ACM (JACM)*, 14(3):543–548, 1967.

[58] Ankan Saha and Ambuj Tewari. On the finite time convergence of cyclic coordinate descent methods. *arXiv preprint arXiv:1005.2146*, 2010.

[59] Robert L Strawderman, Martin T Wells, et al. On hierarchical prior specifications and penalized likelihood. In *Contemporary Developments in Bayesian Analysis and Statistical Decision Theory: A Festschrift for William E. Strawderman*, pages 154–180. Institute of Mathematical Statistics, 2012.

[60] Robert L Strawderman, Martin T Wells, Elizabeth D Schifano, et al. Hierarchical bayes, maximum a posteriori estimators, and minimax concave penalized likelihood estimation. *Electronic Journal of Statistics*, 7:973–990, 2013.

[61] Yoshikazu Takada. Stein's positive part estimator and bayes estimator. *Annals of the Institute of Statistical Mathematics*, 31(1):177–183, 1979.

[62] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[63] Ryan J Tibshirani, Jonathan Taylor, et al. Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232, 2012.

[64] Ryan Joseph Tibshirani, Jonathan E Taylor, Emmanuel Jean Candes, and Trevor Hastie. *The solution path of the generalized lasso*. Stanford University, 2011.

[65] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, 2010.

[66] Paul F Velleman and M Agelia Ypelaar. Constructing regressions with controlled features: A method of probing regression performance. *Journal of the American Statistical Association*, 75(372):839–844, 1980.

[67] Hansheng Wang, Guodong Li, and Guohua Jiang. Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355, 2007.

[68] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[69] Nengjun Yi and Shizhong Xu. Bayesian lasso for quantitative trait loci mapping. *Genetics*, 179(2):1045–1055, 2008.

[70] Ming Yuan and Yi Lin. Efficient empirical bayes variable selection and estimation in linear models. *Journal of the American Statistical Association*, 100(472), 2005.

[71] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

[72] Hui Zou, Trevor Hastie, Robert Tibshirani, et al. On the degrees of freedom of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.

[73] Hui Zou and Runze Li. One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509, 2008.